République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 20 Août 1955 – Skikda

Faculté des Sciences

Département d'INFORMATIQUE

THÈSE DE DOCTORAT

Soumis en conformité avec les exigences du diplôme de 3*éme* cycle doctorat

En : Informatique

Option : Systèmes de cognition et computation

Intitulé :

	Méthodes collaboratives de sensibilité au contexte dans les systèmes multi agents mobiles	

Présenté par :

MAMERI SOUMIA

Soutenu publiquement le 19/12/2024, devant le jury composé de :

Membres du jury :

Président : ***Zeghida Djamel***

Examinatrice ***: Bounour Nora***

Examinateur : ***Boulehouache Sofiane***

Encadrant **:**

Pr. ***Smaine Mazouzi***

Co- Encadreur **:**

Dr. Yacine Kissoum

Remerciements

Tout d'abord, je remercie Dieu de m'avoir donné le courage et la foi pour entreprendre ce travail.

Je tiens à exprimer ma sincère gratitude à mon directeur de thèse, le Dr Kissoum Yacine, Maître de Conférences en Informatique, pour ses conseils avisés, son encadrement bienveillant, ainsi que la liberté de recherche qu'il m'a accordée, ont été d'une grande importance tout au long de cette étude. Ses remarques constructives ont considérablement enrichi la qualité de mon travail.

Je souhaite également adresser mes plus sincères remerciements à Monsieur Mazouzi Smaine, qui a généreusement partagé son expertise et son expérience en tant qu'encadreur. Son soutien constant, ses discussions enrichissantes et ses orientations méthodologiques ont été essentiels à l'avancement de ma recherche.

Je tiens également à remercier Monsieur Redjimi Mohamed pour ses précieux conseils, son aide et ses encouragements.

Mes remerciements s'étendent également à toutes les personnes, connues ou inconnues, qui, au cours de rencontres parfois fugaces, m'ont inspiré et fait découvrir le plaisir d'apprendre.

Enfin, j'exprime ma gratitude aux membres du jury, qui ont généreusement accepté d'évaluer mon travail. Je remercie :

- Monsieur Zeghida Djamel,
- Madame Bounour Nora,
- Monsieur Boulehouache Sofiane.

Leur expertise et leurs remarques enrichiront indéniablement la portée de cette recherche.

Dédicace

A ma source de courage, à ceux que j'ai de plus cher :

Ma maman Mon Père.

A mon mari et à toute ma famille : Khaled, Aissa, Sabri,

Bilel, Soumia, Khouloud, Rayen

Dédicace Spéciale à mes enfants : Yazen et zayd,

Loudjaine

Dédicace à mon beau –père et ma belle mére

A mes chères amies Sara et Amel

MAMERI SOUMIA

Résumé

Dans les réseaux véhiculaires, la gestion de la congestion du trafic demeure un problème crucial, particulièrement aux intersections où les embouteillages affectent directement la fluidité de la mobilité urbaine. Les approches traditionnelles reposant sur des algorithmes fixes ou des modèles statiques montrent leurs limites face à la complexité des environnements dynamiques, marqués par des conditions imprévisibles telles que les variations de trafic, les comportements aléatoires des conducteurs, et les perturbations externes comme les conditions météorologiques. Dans ce contexte, il devient impératif de concevoir des systèmes de gestion du trafic capables de réagir et de s'adapter aux conditions changeantes et imprévisibles.

Cette recherche présente deux contributions majeures pour aborder ces défis. La première consiste en l'exploitation de l'outil RENEW (Reference Net Workshop) pour la modélisation de la gestion de la congestion du trafic aux intersections dans les réseaux véhiculaires (VANET). Cette approche repose sur des systèmes multi-agents mobiles capables de représenter et de simuler les comportements des véhicules dans un environnement dynamique et souvent congestionné. En utilisant des agents mobiles pour modéliser le flux de trafic, cette méthode permet d'optimiser les déplacements et de réduire les embouteillages aux intersections. Bien que cette approche se concentre sur la simulation du comportement des systèmes de transport intelligents sans intégrer de règles complexes, elle constitue une base solide pour comprendre et simuler l'évolution du trafic en tenant compte des variations de trafic et des conditions externes. La deuxième contribution va au-delà de la simple modélisation en intégrant les règles JESS avec l'outil RENEW. Cette intégration permet de renforcer la sensibilité contextuelle des agents dans le modèle en leur fournissant une capacité de raisonnement plus avancée. Grâce à l'intégration de JESS, les agents sont désormais capables de prendre des décisions plus raffinées en fonction de l'évolution dynamique de l'environnement, comme les fluctuations du trafic en temps réel ou les comportements imprévisibles des conducteurs. Cela permet aux agents d'adopter une prise de décision autonome, réactive et sensible au contexte, améliorant ainsi l'efficacité globale de la gestion du trafic aux intersections.

Mots clés : Les réseaux dans les réseaux, Système multi agents mobiles, RENEW, réseaux véhiculaires, la sensibilité au contexte, MULAN, JESS

Abstract

In vehicular networks, traffic congestion management remains a critical issue, particularly at intersections where traffic jams directly impact the smooth flow of urban mobility. Traditional approaches based on fixed algorithms or static models show their limitations in the face of the complexity of dynamic environments, characterized by unpredictable conditions such as traffic variations, random driver behaviors, and external disruptions like weather conditions. In this context, it becomes essential to design traffic management systems capable of reacting and adapting to changing and unpredictable conditions.

This research presents two major contributions to address these challenges. The first involves the use of the RENEW (Reference Net Workshop) tool for modeling traffic congestion management at intersections in vehicular networks (VANET). This approach relies on mobile multi-agent systems capable of representing and simulating vehicle behaviors in a dynamic and often congested environment. By using mobile agents to model traffic flow, this method helps optimize travel and reduce congestion at intersections. While this approach focuses on simulating the behavior of intelligent transportation systems without integrating complex rules, it provides a solid foundation for understanding and simulating traffic evolution, taking into account traffic variations and external conditions.

The second contribution goes beyond simple modeling by integrating JESS rules with the RENEW tool. This integration enhances the contextual sensitivity of the agents in the model by providing them with a more advanced reasoning capability. With the integration of JESS, agents can now make more refined decisions based on the dynamic evolution of the environment, such as real-time traffic fluctuations or unpredictable driver behaviors. This enables agents to adopt autonomous, reactive, and context-sensitive decision-making, thus improving the overall effectiveness of traffic management at intersections.

Key words : Nets within Nets, Mobile Multi-Agent Systems, RENEW, Vehicular Networks, Context Sensitivity, MULAN, JESS

ملخـص

في شبكات المركبات، تظل إدارة الازدحام المروري مشكلة بالغة الأهمية، خاصة عند التقاطعات حيث تؤثر الاختناقات المرورية بشكل مباشر على سيولة التنقل في المناطق الحضرية. تُظهر الأساليب التقليدية القائمة على خوارزميات ثابتة أو نماذج ثابتة حدودها في مواجهة تعقيد البيئات الديناميكية، التي تتميز بظروف لا يمكن التنبؤ بها مثل تغيرات حركة المرور، وسلوك السائق العشوائي، والاضطرابات الخارجية مثل الظروف الجوية. وفي هذا السياق، يصبح من الضروري تصميم أنظمة إدارة حركة المرور قادرة على الاستجابة والتكيف مع الظروف المتغيرة وغير المتوقعة.

يقدم هذا البحث مساهمتين رئيسيتين لمواجهة هذه التحديات. الأول يتكون من استغلال أداة RENEW (Reference Net Workshop) لنمذجة إدارة الازدحام المروري عند التقاطعات في شبكات المركبات (VANET) . يعتمد هذا النهج على أنظمة متنقلة متعددة الوكلاء قادرة على تمثيل ومحاكاة سلوكيات المركبات في بيئة ديناميكية ومزدحمة في كثير من الأحيان. من خلال استخدام وكلاء الهاتف المحمول لنمذجة تدفق حركة المرور، تساعد هذه الطريقة على تحسين السفر وتقليل الاختناقات المرورية عند التقاطعات. وعلى الرغم من أن هذا النهج يركز على محاكاة سلوك أنظمة النقل الذكية دون دمج القواعد المعقدة، إلا أنه يوفر أساسًا متينًا لفهم ومحاكاة تطور حركة المرور مع مراعاة التغيرات المرورية والظروف الخارجية. تتجاوز المساهمة الثانية النمذجة البسيطة من خلال دمج قواعد JESS مع أداة RENEW . يعزز هذا التكامل الحساسية السياقية للوكلاء في النموذج من خلال تزويدهم بقدرة تفكير أكثر تقدمًا. ومن خلال تكامل JESS ، أصبح الوكلاء الآن قادرين على اتخاذ قرارات أكثر دقة بناءً على التغيرات الديناميكية في البيئة، مثل تقلبات حركة المرور في الوقت الفعلي أو سلوكيات السائق غير المتوقعة. وهذا يمكّن الضباط من اعتماد اتخاذ قرارات مستقلة وسريعة الاستجابة ومدركة للسياق، وبالتالي تحسين الكفاءة العامة لإدارة حركة المرور عند التقاطعات.

الكلمات المفتاحية:
الشبكات داخل الشبكات، نظام متنقل متعدد الوكلاء، التجديد، شبكات المركبات، حساسية السياق، MULAN ، JESS .

Table des matières

Introduction Générale **17**

Cadre de travail 17

Motivation 17

Problématiques 18

Objectif 18

Méthodologie 20

1 Sensibilité Contextuelle dans les Systèmes Multi-Agents Mobiles **21**

1.1 Introduction 21

1.2 Le contexte 22

1.2.1 Caractéristiques du contexte 22

1.2.2 Classement du contexte 23

1.3 Sensibilité au contexte 24

1.3.1 Architecture générale d'un système sensible au contexte 25

1.3.1.1 Acquisition de Contexte via les Capteurs 25

1.3.1.2 Interprétation de contexte 26

1.3.1.3 Gestion du contexte 26

1.3.1.4 Adaptation au contexte 27

1.4 Introduction aux systèmes multi-agents 28

1.4.1 Concept d'agent 28

1.4.2 Pourquoi une modélisation basée sur les agents ? 29

1.4.3 Modélisation de systèmes véhiculaires basée sur des agents 30

1.4.4 Systèmes multi-agents . . . 31
1.4.5 L'agent mobile . . . 31
1.4.5.1 Les composants d'un modèle à agents mobiles . . . 31
1.4.5.2 La création et la mort des agents mobiles . . . 32
1.4.5.3 Classification . . . 33
1.4.5.4 Avantages d'agents mobiles . . . 33
1.4.6 Approches Actuelles de Sensibilité au Contexte dans les Systèmes Multi-Agents Mobiles . . . 34
1.4.6.1 Modèle CAMAP de Planification Multi-Agents Sensible au Contexte . . . 34
1.4.6.2 Architecture hybride pour Soutenir les Systemes Sensibles au contexte . . . 34
1.4.6.3 Système Sensible au Contexte pour les Systèmes Cyber-Physiques 36
1.4.6.4 Modélisation Bigraphique pour les Systèmes Contextuels . . . 36
1.4.6.5 Système d'Eclairage Intelligent basé sur des Réseaux de Petri colorés 37
1.4.6.6 Cadre pour les Systèmes Multi-Agents Sensibles au Contexte . . 37
1.5 Conclusion . . . 37

2 Extensions des Réseaux de Petri et le Concept des Réseaux dans les réseaux 39
2.1 Introduction . . . 39
2.2 Les réseaux de petri . . . 39
2.2.1 Concepts de base . . . 40
2.2.2 Définition informelle . . . 41
2.2.3 Définition formelle . . . 42
2.2.4 Propriétés des RdP . . . 43
2.2.5 Avantages du RdP : . . . 45
2.2.6 Les méthodes d'analyse des réseaux de Petri . . . 46
2.2.7 Quelques extensions des réseaux de Petri . . . 46
2.2.7.1 Réseaux de Petri colorés . . . 47
2.2.7.2 Réseaux de Petri temporisés . . . 47
2.2.7.3 Réseaux de Petri synchronisés . . . 48
2.3 Le paradigme (réseaux dans les réseaux) . . . 48

2.3.1 Les réseaux de référence 49
2.3.2 Modélisation de la mobilité 50
2.3.2.1 Types de mobilité 51
2.3.3 Multi Agent Architecture MULAN 54
2.4 Outils de modélisation des réseaux de Petri 56
2.5 Conclusion 56

3 Les réseaux véhiculaires (VANET) 57
3.1 Introduction 57
3.2 Architecture du réseau VANET 57
3.2.1 Modes de communication 58
3.2.1.1 Communication en mode Ad hoc (V2V) 59
3.2.1.2 Communication en mode infrastructure (V2I) 59
3.2.1.3 Communication en mode hybride 60
3.2.2 Types de messages 61
3.2.2.1 Messages de contrôle 61
3.2.2.2 Messages d'alerte 61
3.2.3 Les caractéristiques des VANETS 61
3.2.4 Les services offerts par les réseaux VANET 62
3.2.5 Les applications des VANETS 62
3.2.6 Classification des protocoles de routage dans les réseaux VANET 64
3.2.7 Exploration des Approches SMA dans les Réseaux Véhiculaires : 68
3.3 Conclusion 69

4 Modélisation par agents mobiles pour atténuer la congestion dans les réseaux véhiculaires 70
4.1 Introduction 70
4.1.1 Motivation 70
4.1.2 Objectifs de l'approche 71
4.2 Description générale de l'approche proposée 72
4.2.1 Rôles d'agents 72
4.2.2 Interactions avec les agents 73
4.2.3 Modélisation du système multi-agent proposé 74

4.2.4 SIMULATION 80
4.2.5 Résultats 81
4.2.6 Analyse 83
4.3 CONCLUSION 85

5 Intégration des Règles JESS avec RENEW pour les Véhicules Intelligents 87
5.1 Introduction 87
5.1.1 Objectif et Motivation du travail 88
5.2 Description de l'approche proposée 88
5.2.1 Un Aperçu Approfondi du Contexte Utilisant JESS dans les Systemes Multi-Agents 90
5.2.2 Modelisation de l'environnement 91
5.2.2.1 "Scénario : Une Conduite Intelligente et Sensible au Contexte avec un Véhicule Connecté" 92
5.2.2.2 Représentation Dynamique des Véhicules et des Positions sur la Route" 93
5.2.2.3 Caractéristiques des Agents et Contextes 95
5.2.3 Simulation 99
5.2.4 Exécution et stockage 100
5.2.5 Discussion 101
5.3 Conclusion 102

Table des figures

1.1 Architecture hybride pour la sensibilité au contexte [16] 35

2.1 RdP non borné 43
2.2 RdP vivant 44
2.3 Exemple d'un RdP avec blocage 45
2.4 Réseau réinitialisable 45
2.5 Un réseau objet contenu dans un réseau système 50
2.6 Un réseau objet contenu dans un réseau ststème 51
2.7 Mouvement Spontané 52
2.8 Mouvement Subjectif : Le réseau d'objets déclenche le mouvement 52
2.9 Transport : Le réseau du système déclenche le mouvement 52
2.10 Mouvement Consensuel 53
2.11 Système multi-agents modélisé avec les réseaux dans les réseaux [57] 55

3.1 Les entités de communication 58
3.2 mode de communication v2v [64]. 59
3.3 Modèles de communication véhicule-infrastructure [53]. 60
3.4 Mode de communication hybride [53]. 60
3.5 réduire les accidents 63
3.6 rapport d'accident 63

4.1 L'architecture proposée 74
4.2 Le réseau global du système. 75
4.3 Les réseaux des capteurs du compteur du véhicule. 75

4.4 Le sous-système de feux de circulation . . . 76
4.5 structure de l'agent d'intersection . . . 77
4.6 Protocole de l'agent d'intersection . . . 78
4.7 La structure des agents véhicules et piétons. . . . 78
4.8 Le protocole des véhicules. . . . 79
4.9 Le protocole du piéton. . . . 80
4.10 Un exemple de simulation. . . . 81
4.11 réduction des embouteillages (méthode STLS, AVT). . . . 83
4.12 réduction des embouteillages (notre approche). . . . 84

5.1 Systèmes d'agents sous forme de réseaux dans des réseaux [44] modifiés . . . 88
5.2 Code Java permettant le stockage des résultats . . . 90
5.3 Le système global . . . 94
5.4 L'agent véhicule . . . 98
5.5 Les protocoles de véhicule . . . 99
5.6 Exemple de simulation . . . 100
5.7 Exécution et stockage . . . 101

Liste des tableaux

1.1 catégorisation d'application sensible au contexte [77] 24

4.1 Temps de réponse en intersection 82
4.2 Temps de réponse en intersection (véhicules venant du nord) 83

5.1 Caractéristiques et Contextes des Agents 96

Liste d'abréviations

AVT	AdaptiveValueTracking
ACL	Agent communication langage
AI	Agent Intersection
AP	Agent Piéton
AV	Agent Vehicule
CAMAP	Context Aware Multi Agent Plannification
CPN	Colored Petri Net
CSCW	Computer-Supporte Coopérative Work
E	Est
JESS	Java Expert Systeme Shell
GPS	Global Positioning System
MANET	Mobile ad hoc Network
M2M	Machine To Machine
MULAN	Multi Agent Network
N	Nord
ORM	Object-relational mapping
Plat1	Platforme1
Plat2	Platforme2
Plat3	Platforme3
RENEW	Reference Network
RdP	Réseau de Pétri
RdPT	Réseau de Petri Temporisé
S	Sud

SMA	Systemes Multi Agent
STLS	Standard Traffic Light System
SysML	System Modeling Language
SWRL	Semantique Web Rule Language
UML	Unified Modeling Language
VANET	Vehicule Agent network
V2V	Vehicule to Vehicule
V2I	Vehicule to Infrastructure

Introduction Générale

Cadre de travail

Les systèmes informatiques ont continué à se développer au cours des dernières décennies. Les possibilités d'utilisation, de mise en œuvre et de déploiement deviennent de plus en plus vastes et complexes. La gestion de l'interactivité et de la mobilité devient de plus en plus importante dans les applications les plus utilisées. L'avènement des systèmes multi-agents (ci-après dénommés SMA) apporte donc une nouvelle dimension au concept de modélisation visant à représenter des applications du monde réel. Ces systèmes peuvent travailler et agir dans un environnement commun pour résoudre de manière coopérative d'éventuels conflits. Les systèmes multi-agents (SMA) doivent également être sensibles au contexte pour répondre efficacement aux besoins des utilisateurs dans différents scénarios d'utilisation.

Motivation

La motivation derrière cette étude est de démontrer comment la prise en compte du contexte peut améliorer la gestion des performances dans des environnements complexes tels que les véhicules intelligents. L'intégration de capteurs et de dispositifs de cartographie dans les véhicules génère une grande quantité de données contextuelles. La capacité des systèmes multi-agents (SMA) à traiter ces données de manière adaptative est cruciale pour la prise de décisions en temps réel et l'amélioration de la fluidité du trafic.

Problématiques

La connaissance du contexte permet aux agents de s'adapter dynamiquement à leur environnement en prenant en compte les informations contextuelles pertinentes. Cette étude examine comment la sensibilité du contexte améliore les performances des systèmes multi-agents mobiles, en se concentrant sur des aspects tels que la reconnaissance, la modélisation, l'adaptation, la communication et la collaboration de la conscience du contexte.

En mettant en évidence ces facteurs, nous visons à mieux comprendre comment les systèmes multi-agents peuvent bénéficier d'une sensibilité contextuelle accrue et relever les défis complexes des applications du monde réel. Dans ce contexte, cette étude examine deux contributions importantes à l'amélioration de la sensibilité contextuelle dans les systèmes multi-agents mobiles.

Nous tentons de résoudre une variété de problèmes, dont problèmes de modélisation et de répartition des connaissances entre différents agents, de cohérence des décisions et des plans d'action, ainsi que des problèmes de communication et d'interaction entre agents.

Objectif

Nous montrons que modéliser et tester le comportement des agents, sa mobilité, sa coordination, sa coopération et son interaction est un défi qui doit être surmonté avec des outils formels. Nous supposons que la modélisation de ces systèmes dynamiques nécessite des outils de modélisation avancés qui prennent en compte divers critères de tels systèmes. Notre discussion met l'accent sur les méthodes formelles comme outils les plus appropriés pour valider les exigences de modélisation.

Il convient de souligner que des techniques formelles ont été utilisées pour garantir un niveau assez élevé d'exactitude, de cohérence et de précision. Ceux-ci sont basés sur des principes mathématiques pour réduire le risque d'incertitude et d'ambiguïté. Lors de la phase de conception d'un logiciel, les méthodes formelles permettent à un langage spécifique d'exprimer avec une grande précision les caractéristiques issues du problème fondamental. Celles-ci peuvent être considérées comme des méthodes intéressantes, puissantes et complémentaires entre elles et avec d'autres méthodes classiques telles que les tests et le débogage. Les méthodes formelles peuvent révéler des ambiguïtés, des faits incompréhensibles et des incertitudes. En fait, elle peut découvrir des défauts de conception dans un délai beaucoup plus court que d'autres méthodes de certification traditionnelles qui nécessitent des étapes coûteuses. Suite à une revue de la littérature, nous montrons

qu'un modèle basé sur les réseaux dans les réseaux (NETS WITHIN NETS) est le mieux adapté pour répondre aux exigences de modélisation et de validation des aspects structurels et comportementaux des systèmes dynamiques. Les réseaux dans les réseaux utilisent une représentation graphique combinée à une notation mathématique qui montre clairement un très haut degré de pouvoir expressif.

Dans le cadre de notre étude sur l'amélioration de la sensibilité contextuelle dans les systèmes multi-agents mobiles (SMA), Nous avons choisi le travail sur les véhicules qui seront également équipés de capteurs et de dispositifs permettant de cartographier l'environnement de navigation. La quantité et la complexité de ces informations, combinées à l'échange entre entités via la communication, contribuent à rendre ce système complexe difficile à modéliser. En effet, la nature décentralisée des échanges et de la prise de décision (intégrée au niveau du véhicule ou de l'infrastructure) nécessite des choix de modélisation spécifiques. Quelle échelle de modélisation choisir ? Comment modéliser l'échange d'informations ? Comment modéliser l'infrastructure ? Comment assurer l'autonomie du système ?

Une considération importante est l'intégration de techniques formelles pour valider les exigences de modélisation. Les objectifs de cette étude sont les suivants :

- Analyser comment la sensibilité contextuelle peut améliorer les performances des systèmes multi-agents (SMA).
- Développer des modèles pour tester le comportement des agents en termes de mobilité, de coordination, de coopération et d'interaction.
- Utiliser des techniques formelles pour valider les exigences de modélisation.
- Appliquer ces concepts dans le domaine des véhicules intelligents pour améliorer la gestion de la mobilité et de l'interactivité.

L'objectif principal de notre travail a été d'explorer l'application de la sensibilité au contexte dans les systèmes multi-agents mobiles, en mettant l'accent sur la gestion de la congestion du trafic. Nous avons proposé deux approches successives, visant à améliorer la fluidité du trafic et la réactivité des systèmes face aux variations contextuelles. La première approche s'est concentrée sur l'optimisation du contrôle du trafic aux intersections, tandis que la deuxième a intégré des systèmes d'inférence basés sur des règles pour renforcer l'adaptabilité aux conditions changeantes.

Méthodologie

La suite de la thèse est organisée comme suit : le chapitre 1 présente les notions de base relatives au concept de la sensibilité au contexte, la définition et les catégories et les caractéristiques de contexte, l'architecture générale d'un système sensible au contexte et les approches actuelles de sensibilité au contexte dans les systèmes multi-agents mobiles.

Le deuxième chapitre décrit les méthodes formelles, en commençant par les réseaux de Petri classiques ainsi que leurs extensions, en se focalisant sur les réseaux dans les réseaux et l'architecture MULAN, particulièrement celles dédiées à la modélisation et à la vérification des systèmes multi-agents.

Le troisième chapitre examine les réseaux VANET en décrivant leur architecture, leurs modes de communication (V2V, V2I, hybride), leurs caractéristiques clés, ainsi que les services offerts, notamment en matière de sécurité et de confort. Il aborde également la classification des protocoles de routage et explore l'intégration des approches SMA dans les réseaux véhiculaires.

Dans le quatrième chapitre, nous proposons des systèmes multi-agents mobiles pour atténuer la congestion au sein des intersections. En explorant les différents aspects de la mobilité des agents, nous cherchons à rendre nos systèmes multi-agents encore plus adaptatifs et robustes dans des environnements en constante évolution.

Le cinquième chapitre présente l'application de notre approche sur les véhicules intelligents. Nous étudions, en particulier, les protocoles d'interaction entre les agents. De plus, ce chapitre inclut l'intégration des règles JESS avec l'outil RENEW, ce qui représente une avancée significative dans la création de systèmes multi-agents sensibles au contexte. Cette intégration permet de concevoir des modèles plus complexes et adaptatifs, montrant comment notre approche peut être mise en œuvre concrètement pour améliorer l'efficacité et la réactivité des systèmes de véhicules intelligents.

Chapitre 1

Sensibilité Contextuelle dans les Systèmes Multi-Agents Mobiles

1.1 Introduction

Il n'y a aucun doute que la sensibilité au contexte représente un domaine de recherche très actif dans l'intelligence ambiante. De plus en plus présents dans les environnements technologiques, ces systèmes sont conçus pour prendre en compte les changements de l'environnement et fournir des réponses personnalisées. La connaissance du contexte est essentielle dans divers domaines, tels que la navigation autonome, les interfaces utilisateur intelligentes et d'autres systèmes multi-agents mobiles.

L'intégration de la capacité à traiter des contextes en constante évolution optimise l'efficacité des performances et des réponses, assurant ainsi une expérience utilisateur améliorée. Lors du développement de ces systèmes, il est crucial d'incorporer des technologies adaptées aux environnements complexes. Ces environnements nécessitent une compréhension approfondie du contexte pour garantir des solutions efficaces.

Dans ce qui suit, nous allons détailler la notion de contexte, ses caractéristiques et sa classification, définir la sensibilisation au contexte, et explorer l'architecture générale commune des systèmes sensibles au contexte. De plus, nous discuterons des résultats de recherche importants dans le domaine des applications contextuelles aux systèmes multi-agents, fournissant ainsi un aperçu des avancées et des défis actuels dans ce domaine passionnant.

1.2 Le contexte

Diverses études visent à élaborer une définition claire et précise du concept de contexte. Nous présentons ici une liste non exhaustive de définitions issues de la littérature, classées chronologiquement [11].

Selon [46], une définition plus précise du contexte est avancée, le définissant comme l'ensemble de toutes les entités qui influent sur le comportement cognitif d'un individu (ou d'un système) à un instant donné.

Dans [61], le contexte est considéré comme un concept spécifique défini par l'entité qui le perçoit. Pour une entité, le contexte peut englober sa localisation spatiale, tandis que pour d'autres, il pourrait se référer à sa localisation temporelle ou à son état émotionnel.

Une perspective plus générale du contexte est proposée dans [4], qui est la définition la plus fréquemment adoptée par les concepteurs de systèmes sensibles au contexte. Cette perspective suppose qu'il s'agit de la compilation de toutes les informations pouvant décrire la situation d'une entité.

Une définition du contexte est donnée comme toute information dont le changement de valeur déclenche un service ou modifie la qualité (forme) du service [55].

1.2.1 Caractéristiques du contexte

Le contexte peut changer au fil du temp, de manière dynamique : Les informations contextuelles changent continuellement de valeur avec le temps. Par exemple, le déplacement d'un utilisateur implique que le contexte de localisation de l'utilisateur change de valeurs (par ex. : coordonnées GPS) [46].

Les informations contextuelles sont incohérentes : cette incohérence provient d'informations contextuelles collectées à partir de diverses sources [46]. Concrètement, le contexte peut être constitué d'informations collectées directement à partir de capteurs physiques, récupérées par l'utilisateur à partir de composants logiciels ou de données, ou agrégées à partir de plusieurs sources ou interprétations d'une seule ressource.

Informations contextuelles imparfaites : cette caractéristique du contexte est déterminée par la source d'information. Le contexte peut ne pas être clair si des ressources distinctes fournissent la même information avec différents niveaux de détail ; le contexte peut alors être

imprécis, erroné ou inconnu [35].

Informations contextuelles interdépendantes : les informations contextuelles peuvent dépendre d'autres informations contextuelles. La modification de la valeur d'une information de contexte peut affecter d'autres valeurs contextuelles [35]

1.2.2 Classement du contexte

Pour faciliter la collecte et la présentation d'informations contextuelles dans un système sensible au contexte, une catégorisation est nécessaire. Certains travaux ont proposé une classification contextuelle. [43] classe les contextes en deux types : le contexte physique de l'utilisateur et le contexte organisationnel. Le contexte physique inclut les paramètres utilisateur tels que les caractéristiques de l'emplacement, du matériel et des applications. Le contexte organisationnel représente les informations liées aux processus collaboratifs auxquels les utilisateurs participent, tels que les fonctions et les calendriers.

Selon [83], le contexte est divisé en trois axes : le contexte physique, le contexte utilisateur et le contexte social. Le contexte physique est représenté par les caractéristiques physiques du périphérique, telles que la bande passante. Le contexte utilisateur regroupe les préférences de l'utilisateur. Le contexte social apparaît dans le cas d'interactions entre utilisateurs existants et nouveaux. La combinaison des deux classifications fournit des informations contextuelles qui peuvent être classées selon le contexte utilisateur, le contexte spatio-temporel, le contexte physique et le contexte organisationnel [93] :

Contexte utilisateur : décrit les préférences de l'utilisateur, telles que la langue, la résolution d'écran, etc. Ces informations sont souvent imposées par les utilisateurs eux-mêmes.

Contexte physique : comprend les paramètres physiques de l'environnement de travail. Cette catégorie décrit les caractéristiques du terminal (mémoire disponible, vitesse de traitement, etc.) et du réseau de connexion (bande passante, mode de connexion, canal de connexion). D'autres paramètres physiques peuvent également être inclus, tels que la température.

Contexte organisationnel : comprend des informations pertinentes à la collaboration à laquelle un utilisateur participe, comme le groupe auquel il appartient, son rôle, ses activités en cours, les ressources partagées entre les membres du groupe, les autorisations possibles, etc.

Une approche similaire, fondée sur le service, est appliquée et propose de classer les informations contextuelles en deux classes [55] Celles-ci incluent :

Les informations d'activation du service : un ensemble d'informations dont la valeur change et active automatiquement le service fourni par l'appareil.

Les informations de changement de forme d'un service : regroupent les informations dont le changement de valeur provoque le changement de forme d'un service (qualité).

1.3 Sensibilité au contexte

Après avoir exploré diverses définitions du concept de contexte, notre attention se tourne désormais vers les définitions associées à la sensibilité contextuelle.

Selon [73], la sensibilité contextuelle réside dans la capacité d'un système à percevoir et à réagir aux informations de son environnement, telles que l'heure, la température, la localisation ou l'identité d'une personne.

D'après [4], une application est considérée comme sensible au contexte si elle est apte à discerner son contexte et à guider son comportement dynamique en fonction de la connaissance de son environnement.

Selon Ferrand et Onaindia [27], les systèmes ont été conçus pour être sensibles au contexte, c'est-à-dire qu'ils peuvent détecter et interpréter le contexte de l'environnement, comme la présence de personnes, d'objets et d'événements, et utiliser ces informations pour fournir des services proactifs et personnalisés.

Ensuite, les applications sensibles au contexte sont classifiées en quatre catégories, basées sur deux critères : la réalisation manuelle ou automatique des tâches, et le type de tâche, c'est-à-dire obtenir des informations ou réaliser des actions [76] (voir tableau 1).

	Manuelle	Automatique
Information	Selection à proximité	Reconfiguration contextuelle automatique
Commande	Commande contextuelle	Action contextuellement déclenchées

TABLE 1.1 – catégorisation d'application sensible au contexte [77]

Une autre perspective, présentée par [53], stipule que la sensibilité au contexte se manifeste lorsqu'un système est capable d'extraire, d'interpréter et d'utiliser des informations contextuelles, en ajustant ainsi ses fonctionnalités en fonction du contexte d'utilisation actuel.

En parallèle, une définition similaire est avancée par [76, 18], soulignant que la sensibilité au contexte se caractérise par l'autonomie dans la détection et la réponse aux variables environnementales.

Les définitions qui englobent divers axes de sensibilité contextuelle permettent de dégager deux aspects fondamentaux de cette caractéristique : la capacité à percevoir son contexte et la capacité à exploiter le contexte perçu. Ces deux aspects se reflètent clairement dans la définition de [53], qui définit la sensibilité contextuelle comme "la capacité à percevoir, interpréter et utiliser les différentes informations relatives au contexte courant pour adapter dynamiquement sa fonctionnalité".

1.3.1 Architecture générale d'un système sensible au contexte

De nombreuses recherches ont élaboré des architectures pour les systèmes sensibles au contexte [56, 85, 9], qui expose la configuration globale d'un système contextuel. Cette structure se décompose en quatre niveaux : Acquisition du contexte, Interprétation du contexte, Gestion du contexte et Adaptation au contexte.

Chacune de ces couches joue un rôle dans le système, dans lequel nous présenterons les couches de l'extérieur vers l'intérieur avec le rôle de chaque couche.

1.3.1.1 Acquisition de Contexte via les Capteurs

Les systèmes contextuels sont spécifiquement conçus pour s'adapter aux variations de l'environnement qui les entoure. Le processus fondamental de ces applications sensibles au contexte commence par la perception et la collecte d'informations contextuelles. Ces informations peuvent provenir de diverses sources, telles que des capteurs en temps réel ou des données extraites d'une base de connaissances, englobant des éléments comme la localisation GPS, la température, l'humidité, la concentration de polluants, ainsi que des informations personnelles et des données historiques des utilisateurs.

La couche d'acquisition du contexte est composée d'un ensemble de capteurs, classés en capteurs physiques et logiques. Pour garantir une collecte de données contextuelles efficace, le système doit transformer ces données pour les rendre compréhensibles. Cette transformation représente une forme d'interprétation fournie par la couche d'interprétation du contexte [56].

1.3.1.2 Interprétation de contexte

Le rôle de cette couche est d'interpréter les données contextuelles fournies par les capteurs. Elle analyse et transforme les données brutes fournies par la couche capteur ; des opérations sont effectuées sur l'extraction, la quantification et la synthèse des données brutes. Cette couche peut également jouer un rôle dans la résolution des conflits provoqués par l'utilisation de certaines ressources contextuelles. Ces sources peuvent donner des résultats contradictoires et conduire à des situations incorrectes. Cette couche doit donc avoir les moyens de résoudre ces conflits [85].

1.3.1.3 Gestion du contexte

Après avoir collecté et interprété les données contextuelles pour les rendre compréhensibles par le système, la couche de gestion du contexte prend en charge la représentation formelle et le stockage de ces données [55]. La représentation et la modélisation du contexte constituent la base essentielle pour obtenir un contexte de niveau supérieur. La structuration des informations contextuelles est réalisée en fonction d'un modèle spécifique, et le choix de ce modèle dépend étroitement des mécanismes choisis pour adapter le fonctionnement du système au contexte.

Nous examinerons dans la suite les méthodes de modélisation du contexte les plus adaptées à cette tâche.

Méthodes de Modélisation de Contexte

✓ **Modèles graphiques**

- Les modèles graphiques tels qu'UML et l'Object Role Model (ORM) [34] sont considérés comme appropriés pour la modélisation du contexte.
- Le diagramme de contexte système, similaire au diagramme de cas d'utilisation UML, est recommandé pour visualiser l'ensemble du système, y compris les entrées et sorties provenant des facteurs externes et les interactions avec eux [34].

✓ **SysML (System Modeling langage) :**

- Les diagrammes de définition de blocs en SysML sont décrits comme des diagrammes de contexte système. Ils fournissent une vue d'ensemble des interactions du système avec son environnement, en se concentrant sur les frontières du système et sur la manière dont il communique avec les entités externes [54].

- Les annotations UML, avec des annotations [10, 85], sont utilisées pour regrouper les éléments de modèle de différents diagrammes et spécifier les informations pertinentes liées à un contexte particulier. Les diagrammes de cas d'utilisation sont souvent appliqués pour différencier un système de son environnement.

✓ **Modélisation Orientée Aspect**

Certains travaux utilisent la modélisation orientée aspect, où l'aspect du contexte nécessite le comportement de composants qui ne sont pas dans le modèle de domaine.

✓ **Modélisation Basée sur la Logique**

Une approche basée sur la logique [15] décrit le contexte par des événements, des prédictions ou des rôles, avec pour objectif la création de nouvelles expressions ou faits à partir d'expressions précédentes.

✓ **Réseau de Petri et Réseau de Petri Coloré**

Les réseaux de Petri, ainsi que les réseaux de Petri colorés [40, 31, 32], sont appliqués à la modélisation du contexte en raison de leur capacité à représenter des systèmes dynamiques, des systèmes à événements discrets et des systèmes concurrents. Les places du réseau de Petri correspondent à des situations contextuelles, les transitions représentent des relations entre ces situations, et les événements sont liés aux transitions, symbolisant des changements d'état.

D'autres approches, telles que l'utilisation de réseaux de neurones, de machines à vecteurs de support, de réseaux bayésiens et de logique floue, ont également été employées dans le domaine des systèmes sensibles au contexte.

En résumé, le choix de la méthode de modélisation dépend des caractéristiques spécifiques du contexte, des besoins du système et des mécanismes d'adaptation prévus. Chaque approche offre des avantages particuliers en fonction du contexte d'application.

1.3.1.4 Adaptation au contexte

L'adaptation au contexte englobe un ensemble de réponses planifiées en réaction aux variations du contexte, reposant sur des règles spécifiques. Ces règles sont généralement mises en œuvre à l'aide de langages de programmation traditionnels ou de logiques de prédicat. Pour atténuer l'incertitude liée au contexte, certaines recherches se tournent vers l'utilisation de logiques floues ou probabilistes. Diverses formes d'adaptation peuvent être identifiées [66].

Adaptation Architecturale : Cette forme d'adaptation spécifie les ajustements en temps réel apportés à la structure des composants du système et/ou à leurs interactions, en se basant sur le modèle architectural du système.

Ajustements des Ingrédients : Il s'agit de modifier la structure et le comportement d'une entité logicielle en réponse aux changements survenant dans son environnement d'exécution, et ce, en temps réel.

Ajuster la Structure : L'adaptation structurelle implique le changement dynamique du type de composant d'application, tels que les signatures de méthode.

Adaptation Comportementale : Elle se réfère aux changements dynamiques intervenant pendant la phase d'exécution d'un composant logiciel, que ce soit en modifiant sa configuration ou en interceptant ses requêtes et réponses.

Ajuster le Contenu : Cette forme d'adaptation implique la transformation et la manipulation de contenus en fonction des caractéristiques de l'application et du terminal utilisé.

Ces différentes formes d'adaptation illustrent la diversité des approches visant à répondre de manière flexible aux variations du contexte, en ajustant tant la structure que le comportement des composants logiciels en fonction des besoins et des conditions du moment.

En examinant ces diverses formes d'adaptation, il est clair que l'adaptation joue un rôle crucial dans la conception et le fonctionnement des systèmes sensibles au contexte. En effet, elle représente la dernière couche de leur architecture, permettant aux systèmes de s'ajuster en temps réel aux changements environnementaux. Cette capacité d'adaptation est d'une importance capitale dans les systèmes multi-agents, où les entités autonomes doivent interagir et collaborer de manière efficace dans des environnements dynamiques et complexes. Les principes d'adaptation présentés ici fournissent une base solide pour explorer comment les agents dans un tel système peuvent ajuster leur comportement, leur structure et leur interaction en réponse aux fluctuations du contexte.

1.4 Introduction aux systèmes multi-agents

1.4.1 Concept d'agent

Un agent est toute entité physique ou logique capable de percevoir son environnement par l'intermédiaire de capteurs et d'agir sur cet environnement de manière autonome grâce à ses effecteurs [72].

Les travaux de Ferber [26, 27], qui font référence dans ce domaine, convergent vers une définition plus étendue d'un agent. Un agent est une entité physique ou virtuelle :

- capable d'agir dans un environnement,
- pouvant communiquer directement avec d'autres agents,
- animée par un ensemble de tendances (objectifs, satisfaction, etc.),
- possédant ses propres ressources,
- dotée d'une capacité de perception de son environnement,
- ayant une connaissance partielle et locale de son environnement.
- disposant de compétences et pouvant offrir des services.
- pouvant éventuellement se reproduire.
- dont le comportement vise à satisfaire ses objectifs en fonction des ressources et compétences dont il dispose, de sa perception, de ses représentations et des communications qu'elle reçoit.

1.4.2 Pourquoi une modélisation basée sur les agents ?

Nous vivons dans un monde très complexe où de multiples éléments interagissent, communiquent, négocient et travaillent ensemble pour atteindre des objectifs. Cependant, la modélisation basée sur les agents est utilisée pour les raisons suivantes :

• • les systèmes qui doivent être modélisés et simulés sont de plus en plus complexes en termes d'interdépendances et d'interactions, et (ii) certains systèmes seront toujours trop complexes pour être abordés par des méthodes simples. Modélisation basée sur des règles, modélisation d'équations, etc. Par rapport à d'autres techniques de modélisation, la technologie des agents présente plusieurs caractéristiques et de nombreux avantages qui permettent la modélisation et la simulation de systèmes complexes. En fait, les caractéristiques les plus importantes de la modélisation basée sur les agents sont les suivantes :

- Sans programmation explicite, la modélisation basée sur les agents permet de créer des phénomènes spécifiques résultant d'interactions entre agents lorsqu'ils communiquent et collaborent pour accomplir des tâches collectives.
- La modélisation basée sur les agents fournit un modèle qui ressemble à la description et au comportement du système modélisé.

- La modélisation basée sur les agents permet d'ajouter de nouveaux agents au modèle final, créant ainsi un modèle flexible. Cette caractéristique détermine le comportement de chaque agent (niveau de rationalité, capacité à apprendre, mesure des performances, etc.).
- La modélisation basée sur les agents se caractérise par un processus flexible qui permet de formaliser le comportement et les interactions de chaque agent à l'aide d'équations ou de règles de décision si-alors.

1.4.3 Modélisation de systèmes véhiculaires basée sur des agents

La modélisation et la simulation basées sur des agents sont utilisées pour concevoir et implémenter de nombreux systèmes, en résolvant divers problèmes et en offrant différents services. En particulier, cette approche est adoptée pour décrire et simuler le comportement de nombreux systèmes véhiculaires, notamment :

- **Modélisation du flux de trafic :**

  La modélisation du comportement des conducteurs humains à l'aide d'agents, en particulier les processus de contrôle longitudinal et latéral, permet d'analyser l'impact de divers paramètres (par exemple, le style de conduite, le temps de réaction du conducteur, etc.) sur le flux de trafic [19, 53].

- **Contrôle du flux de trafic :**

  Pour minimiser le temps d'attente des usagers de la route et ainsi augmenter le flux de trafic, de nombreux systèmes de contrôle de trafic ont été modélisés et simulés en utilisant des agents [3, 8].

- **Systèmes d'aide à la conduite :**

  Ces systèmes aident les conducteurs humains pour améliorer la sécurité routière. Plusieurs systèmes, tels que les systèmes anticollision et de navigation, ont été proposés et modélisés à l'aide de la modélisation basée sur des agents [3, 8].

- **Simulation microscopique :**

  Utilisée pour tester et valider des hypothèses ainsi que pour évaluer la performance des systèmes véhiculaires proposés. Dans ce type de simulation, un agent contrôle un véhicule en implémentant des modèles de suivi des véhicules et de changement de voie [94, 80].

1.4.4 Systèmes multi-agents

Selon la définition de Demazeau [94], les systèmes multi-agents (SMA) présentent les caractéristiques suivantes :

Environnement E : Espace avec métriques et support de communication.

Ensemble O d'objets : Ces objets sont dans l'environnement E. Il est donc possible de : Mapper O aux coordonnées de E à tout moment.

Ensemble d'agents A : les unités autonomes actives du système.

Séquence d'interactions I : Les agents A sont capables de modifier l'environnement E.

Divers opérateurs opérationnels : peuvent agir sur l'environnement et provoquer des réactions.

1.4.5 L'agent mobile

L'agent mobile n'est pas lié au système à partir duquel il initie l'exécution. Les agents mobiles peuvent se déplacer d'un hôte à un autre au sein du réseau. Il peut transporter son état et son code d'un environnement à un autre dans le réseau où il continue de s'exécuter [25].

Les agents mobiles ont suscité un intérêt considérable ces dernières années car ils permettent des interactions asynchrones et peuvent réduire le trafic réseau lors des interactions avec les clients.

1.4.5.1 Les composants d'un modèle à agents mobiles

Les deux concepts de base du modèle d'agent mobile sont l'agent et son environnement appelé système d'agents [36].

L'agent mobile est une entité dotée de cinq propriétés : son état, sa mise en œuvre, son interface, son identifiant et son autorité. Lorsqu'un agent se déplace dans le réseau, il transporte avec lui ses attributs.

Système d'agents (également appelé serveur d'agent) est un environnement capable de créer, d'interpréter, d'exécuter, de valider et d'arrêter un agent. Comme un agent, un système d'agents est lié à une autorité qui identifie l'individu ou l'entité pour lequel il agit. Un système d'agent est identifié par son nom et son adresse. Un serveur peut contenir plusieurs systèmes d'agent.

Cinq concepts jouent un rôle important dans le système de system d'agent : La place, Le type d'un système d'agent, Les ressources, Localisation, Région.

1.4.5.2 La création et la mort des agents mobiles

✓ Les agents sont créés au même endroit. La création peut être initiée par un autre agent au même emplacement ou par un système agent ou non-agent en dehors de cet emplacement. L'appelant doit s'authentifier auprès du site et établir les autorisations et les informations d'identification dont dispose l'agent. Les créateurs peuvent également définir des arguments d'initialisation pour l'agent. Les classes de définition requises pour instancier un agent peuvent résider sur l'hôte local ou sur une machine distante, ou être fournies par l'auteur.

La création se déroule en trois étapes :

✓ **Instancier et attribuer des identifiants :** le code de la classe d'agent est chargé et exécuté. Un objet agent est instancié. L'emplacement attribue un identifiant unique à un agent.

✓ **Initialisation :** un agent peut s'initialiser à l'aide des arguments d'initialisation fournis par son créateur. Une fois l'initialisation terminée, l'agent est entièrement installé.

✓ **Exécution autonome :** après l'installation, l'agent peut commencer à s'exécuter. Il peut désormais fonctionner indépendamment des autres agents colocalisés. Dans la plupart des cas, les agents sont supprimés en quittant les lieux. Ce processus peut être initié par l'agent lui-même ou par un autre système agent ou non-agent, qu'il soit colocalisé ou non. Les agents peuvent également être démis de leurs fonctions par le système pour plusieurs raisons suivantes :

— La fin de son « temps de vie ».

— L'agent n'est ni utilisé ni référencé.

— Violation des règles de sécurité.

— Le système est éteint.

— Le processus de destruction commence en deux étapes :

— Préparation : une opportunité est donnée à l'agent de terminer sa tâche actuelle avant d'être supprimé.

— Suspension de l'exécution" pour correspondre au contexte : où l'activité de l'agent est suspendue.

1.4.5.3 Classification

Les agents mobiles ont suscité un intérêt considérable ces dernières années, car ils prennent en charge les interactions asynchrones et peuvent réduire le trafic réseau lors des interactions avec les clients.

1.4.5.4 Avantages d'agents mobiles

Les avantages potentiels et courants de l'utilisation d'agents mobiles dans le réseau sont [25] :

- ✓ **Adaptation dynamique :** les agents mobiles peuvent surveiller l'environnement et réagir de manière indépendante, afin d'ajuster leur comportement futur en fonction des informations collectées.
- ✓ **Operations en environnement hétérogène :** les agents mobiles dépendent uniquement de leur environnement d'exécution, les réseaux étant de plus en plus hétérogènes, tant en termes matériels que de logiciels, l'utilisation d'agents mobiles peut faciliter l'interaction entre différents appareils
- ✓ **Réduire la charge du réseau :** L'interaction des protocoles de communication crée souvent des taux de trafic élevés dans le réseau. Avec les agents mobiles, les interactions peuvent être effectuées localement. De plus, déplacer le code là où résident les données peut bénéficier aux réseaux qui traitent de grandes quantités de données.
- ✓ **Exécuter de manière asynchrone et indépendante :** Les appareils mobiles utilisant la communication sans fil ont des connexions instables, ce qui rend difficile, voire impossible, l'exécution de tâches nécessitant une connexion continue. Après avoir été envoyée à un serveur distant, une tâche Les agents peuvent fonctionner de manière asynchrone et indépendante ; la connexion peut être rétablie ultérieurement pour restaurer l'agent.
- ✓ **Réduire la latence :** dans les systèmes en temps réel, où les retards dans la prise de décision sont souvent inacceptables, des agents mobiles peuvent être dépêchés pour agir localement et effectuer rapidement des actions de contrôle.

1.4.6 Approches Actuelles de Sensibilité au Contexte dans les Systèmes Multi-Agents Mobiles

Le domaine des systèmes contextuels est actuellement au cœur de la recherche, et les systèmes multi-agents y jouent un rôle central. Cette section se concentre sur certaines approches notables de systèmes multi-agents sensibles au contexte, illustrant la diversité des solutions proposées.

1.4.6.1 Modèle CAMAP de Planification Multi-Agents Sensible au Contexte

[27] L'implémentation et l'évaluation d'un modèle de Planification Multi-Agent Sensible au Contexte (CAMAP), particulièrement appliqué aux scénarios de soins de santé dans des environnements AmI, utilisant un type spécial d'argumentation, connu sous le nom d'argumentation réfutable [36]. Plus précisément :

a) CAMAP aborde les problèmes de planification dans les environnements AmI où plusieurs agents ambiants extraient des données contextuelles brutes, font des inférences contextuelles et coopèrent dans le processus de planification.

b) Le modèle peut être utilisé dans de nombreuses applications réelles où les capacités de perception du contexte et de planification sont distribuées entre les agents ambiants.

c) CAMAP est capable de résoudre des problèmes avec différents niveaux de complexité et de couplage (faiblement couplés et fortement couplés).

d) Les agents impliqués dans le problème expriment leurs avis sur la construction du plan en utilisant l'argumentation réfutable. Cela permet aux agents d'interagir progressivement et de discuter de l'impact et des conséquences des informations contextuelles sur les actions du plan et de présenter des arguments pour ou contre un choix d'action particulier.

e) Les agents sont capables de choisir la meilleure action pour le plan en cours de construction.

1.4.6.2 Architecture hybride pour Soutenir les Systemes Sensibles au contexte

Une architecture pour un environnement contextuel a été proposée, comprenant cinq couches interconnectées, comme le montre la figure suivante [16] .

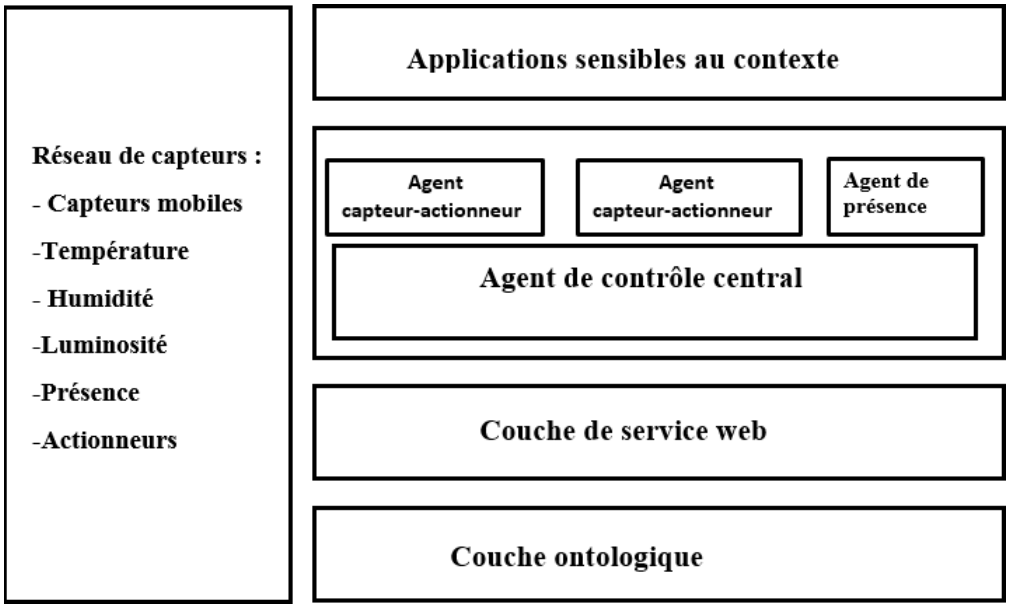


FIGURE 1.1 – Architecture hybride pour la sensibilité au contexte [16]

- **Réseau de capteurs :** Cette couche collecte des données environnementales via des capteurs physiques et mobiles, surveillant des variables telles que la température, l'humidité et la présence humaine.
- **Agents intelligents :** Ces programmes autonomes, intégrés dans des microcontrôleurs, interagissent avec les capteurs pour surveiller et contrôler l'environnement physique. Ils comprennent des agents de capteurs et d'actionneurs, un agent de contrôle central et un serveur de contexte.Cette couche collecte des données environnementales via des capteurs physiques et mobiles, surveillant des variables telles que la température, l'humidité et la présence humaine.
- **Ontologies pour la modélisation du contexte :** Des modèles basés sur des ontologies représentent les données contextuelles, permettant une représentation sémantique formelle et facilitant l'inférence sur les données stockées.
- **Services Web :** Ces services gèrent les données, effectuent l'extraction et le stockage d'informations, ainsi que la vérification de la cohérence des données, en utilisant les ontologies comme base.
- **Applications sensibles au contexte :** Ces applications mobiles permettent l'interaction avec les utilisateurs, offrant des fonctionnalités telles que la récupération d'informations, les interfaces vocales et en langage naturel, et la présentation d'informations pertinentes et op-

portunes.

Cette architecture vise à créer un environnement intelligent et adaptable, où les actions sont déclenchées en fonction du contexte actuel, améliorant ainsi l'expérience utilisateur et soutenant la prise de décision.

1.4.6.3 Système Sensible au Contexte pour les Systèmes Cyber-Physiques

Les auteurs ont développé des systèmes médicaux cyber-physiques (MCPS) pour améliorer les soins de santé et réduire les coûts. Ils ont introduit un détecteur appelé Parameter-Invariant (PAIN) pour détecter en temps réel les événements critiques chez les patients, mais celui-ci reste vulnérable aux données incorrectes ou manquantes. Pour résoudre ce problème, ils ont proposé d'incorporer des données contextuelles dans le détecteur PAIN afin de réduire les fausses alarmes. En utilisant ces données supplémentaires, ils ont cherché à améliorer la fiabilité du détecteur et à réduire les alarmes inutiles dans les environnements hospitaliers [38].

1.4.6.4 Modélisation Bigraphique pour les Systèmes Contextuels

Ils ont réalisé dans cette étude [80] :

- **Extension des bigraphes pour l'IoT :** Ils ont décrit l'extension des bigraphes, notée BiAgents*, pour permettre la spécification des systèmes IoT. Cela inclut la modélisation de la partie physique des systèmes IoT à l'aide des bigraphes pour capturer la localité et la connectivité des composants.
- **Modélisation de la partie Intelligente des Systèmes IoT :** Ils ont adapté les BiAgents pour les systèmes IoT, permettant aux agents d'analyser l'environnement, d'interagir entre eux et de prendre des décisions intelligentes. Cette modélisation inclut un système intelligent de détection de collision comme cas d'étude.
- **Implémentation dans Maude :** Ils ont encodé les spécifications BiAgents* dans le langage Maude pour permettre une exécution automatique des spécifications. Maude a été choisie pour sa simplicité, son expressivité et ses performances en termes de temps d'exécution.
- **Validation et Vérification :** Ils ont utilisé Maude pour valider les spécifications et garantir la vérification formelle de propriétés de sécurité et de correction. Les outils existants basés sur les BRS, tels que BigraphER et BPL Tool, n'étant pas adaptés, ils ont mis en œuvre leur propre spécification dans Maude.

En résumé, cette étude propose un modèle formel et exécutable pour les systèmes IoT basé sur les bigraphes et les agents formels, validé et exécuté à l'aide de Maude pour assurer la qualité, la sécurité et la correction des systèmes IoT complexes [79].

1.4.6.5 Système d'Eclairage Intelligent basé sur des Réseaux de Petri colorés

Un système d'éclairage intelligent basé sur les réseaux de Petri colorés a été proposé pour l'IoT, capable d'ajuster automatiquement les paramètres d'éclairage en fonction des préférences utilisateur, des conditions de luminosité extérieure et de l'heure du jour. Ce système utilise un réseau neuronal artificiel pour calculer les réglages optimaux, offrant ainsi un niveau de confort supplémentaire aux utilisateurs [51].

1.4.6.6 Cadre pour les Systèmes Multi-Agents Sensibles au Contexte

Les auteurs de ce travail ont développé un modèle conceptuel et un prototype de système pour le jardinage intérieur intelligent. Ce système, appelé Planquarium, permet de cultiver des légumes frais et de purifier l'air intérieur sans nécessiter de connaissances spécialisée. Ils ont utilisé une modélisation contextuelle basée sur des ontologies pour créer un environnement intelligent et conscient du contexte. Cela permet une compréhension partagée et une meilleure communication entre les composants du système. Une architecture basée sur les agents a été mise en œuvre pour gérer la communication et l'interopérabilité entre les différents composants du système. Les auteurs ont implémenté les spécifications de leur modèle en utilisant le langage Maude pour permettre une exécution automatique et une vérification formelle des propriétés de sécurité et de correction. En développant des règles et des modèles d'interopérabilité sémantique, ils ont abordé les difficultés liées à la synchronisation des composants du système intelligent. Une analyse détaillée des résultats expérimentaux a été réalisée pour valider l'efficacité et la performance du système proposé [86]. Chacune de ces contributions apporte une perspective unique sur la sensibilité au contexte dans les systèmes multi-agents, mettant en évidence la diversité des approches pour relever les défis complexes de ce domaine de recherche.

1.5 Conclusion

Ce chapitre est consacré aux aspects fondamentaux de la démarche visant une appréhension complète des systèmes sensibles au contexte. Notre exploration a débuté par une mise en lumière

des définitions les plus couramment utilisées pour définir le concept de contexte, en mettant en relief ses caractéristiques intrinsèques et en identifiant divers types de contexte reconnus. Par la suite, nous avons approfondi la notion de sensibilité au contexte, exposant son architecture générale et présentant un échantillon représentatif de travaux liés à cette sensibilité, particulièrement dans le domaine des systèmes multi-agents.

Le prochain chapitre abordera une analyse plus approfondie des réseaux de Petri et plus précisément des réseaux dans des réseaux de l'architecture MULAN. Ces éléments compléteront notre panorama des outils et des méthodologies fondamentaux pour la conception de systèmes adaptatifs et réactifs aux variations de leur environnement.

Chapitre 2

Extensions des Réseaux de Petri et le Concept des Réseaux dans les réseaux

2.1 Introduction

Les réseaux de Petri [62] constituent un paradigme de base pour la modélisation de systèmes discrets (systèmes informatiques, systèmes de fabrication, systèmes de communication, etc.). Ils ont été inventés par Carl Adam Petri et sont utilisés dans de nombreux domaines et notamment pour la modélisation des systèmes de production, de transport et de communication. Dans le contexte de la sensibilité au contexte et des travaux utilisant les SMA (Systèmes Multi-Agents), il est pertinent de souligner que les réseaux de Petri jouent un rôle crucial.

Dans cette section, nous présenterons d'abord le concept général des réseaux de Petri. Ensuite, nous aborderons les systèmes de référence de ces réseaux. Enfin, nous présenterons l'architecture MULAN (Multi-Agent Net architecture).

2.2 Les réseaux de petri

Les réseaux de Petri [62] offrent des fonctionnalités remarquables, telles que la capacité à modéliser et visualiser des comportements parallèles, ainsi que la synchronisation et le partage de ressources. Leur base théorique solide a été largement étudiée, produisant des résultats théoriques riches. Principalement utilisés par les informaticiens et les automaticiens, ces réseaux constituent un outil polyvalent pour la modélisation d'une grande variété de phénomènes. Leur utilisation

s'étend à plusieurs domaines, permettant notamment :

- ✓ La modélisation de systèmes informatiques.
- ✓ L'évaluation des performances des systèmes discrets et des interfaces homme-machine.
- ✓ La gestion des commandes d'usine.
- ✓ La conception de systèmes en temps réel.
- ✓ La modélisation de protocoles de communication.
- ✓ La représentation de la chaîne de production dans le secteur manufacturier.

2.2.1 Concepts de base

- **Condition :** Dans le contexte d'un système, une condition, est une assertion logique sur l'état du système qui peut être vraie, soit fausse. Cette condition est représentée dans le modèle par une place.

- **Evénement :** Les événements représentent les actions qui surviennent dans le système. Leur déclenchement est conditionné par l'état courant du système.

  Un état du système peut être conceptualisé comme une collection de conditions. Cette représentation peut être réalisée au moyen d'une transition dans le modèle.

- **Déclenchement, précondition, post-condition :** Lorsqu'un événement se produit, il est possible que certaines de ses conditions préalables ne soient plus valides, tandis que d'autres conditions, appelées post-conditions de l'événement, deviennent valides. La satisfaction d'une condition est représentée dans le modèle à par des jetons.

Condition vraie

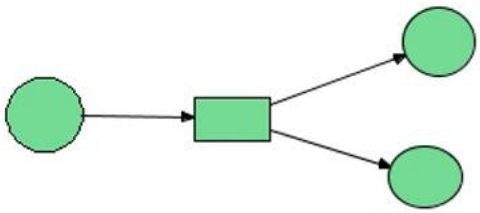
Condition fausse

2.2.2 Définition informelle

De manière informelle, un réseau de Petri est un graphe orienté constitué de deux types de nœuds :

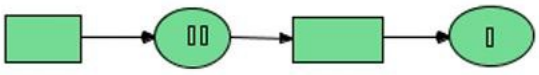
- Les places (Pi), qui décrivent les états du système modélisé, représentées par un ensemble P= P1, P2, ...
- Les transitions (Ti), qui symbolisent les changements d'états du système, représentées par un ensemble T= T1, T2, ...

Les places et les transitions sont connectées par des arcs orientés, formant ainsi un graphe biparti orienté. Chaque arc est associé à un poids (un nombre entier), par défaut égal à 1.



Exemple 1

a. Marquage Dans un réseau de Petri, chaque place peut contenir un nombre entier de marques ou jetons, qui peut être positif ou nul. On utilise généralement M(Pi) ou simplement mi pour représenter le nombre de marques dans une place Pi spécifique. Le marquage du réseau à un instant donné, noté Mi, est défini par un vecteur contenant ces marques, c'est-à-dire Mi = (m1, m2, ..., mn), où chaque élément représente le nombre de marques dans une place spécifique. Le marquage initial, désigné par M0, décrit l'état initial du système modélisé.



Exemple 2

Ce RdP possède 2 places, 2 transitions et 3 arcs orientés. Soit donc : Le marquage initial est M0 = (2,1)

b. Franchissement d'une Transition : Pour représenter l'évolution du système modélisé, les réseaux de Petri utilisent un formalisme permettant de passer d'un état à un autre, appelé franchissement des transitions. Ce processus ne peut se produire que si toutes les places en amont de la transition contiennent un nombre de jetons suffisant, égal ou supérieur au poids des arcs correspondants. On dit alors que la transition est franchissable ou validée. Le franchissement est une opération indivisible qui consiste à retirer des jetons des places en amont et à en ajouter dans les places en aval de la transition franchie. Le nombre de jetons retirés ou ajoutés est déterminé par le poids de l'arc reliant la transition aux places concernées.

2.2.3 Définition formelle

Formellement, un RdP est un quintuple, R= (P, T, F, W, M0) tel que :

- $P = \{p1, p2, ..., pm\}$ Un ensemble de m places, où m est un nombre fini ;
- $T = \{T1, T2, ..., Tn\}$ ensemble fini de n transitions ;
- $F \in (PXT) \cup (TXP)$ ensemble d'arcs ;
- $W : F\{1, 2, 3.....\}$ fonction de poids ;
- $M0 : P\{0, 1, 2, 3.....\}$ marquage initial ;

Un réseau de Petri dont tous les arcs sont pondérés avec le poids "1" est désigné comme un réseau de Petri ordinaire. Lorsque les arcs peuvent avoir des poids supérieurs à "1", le réseau est qualifié de réseau de Petri généralisé. La structure de RdP est donnée par le quadruplet

Q= (P, T, F, W), Q représente le RdP sans aucune spécification de marquage initial M0.

Un RdP marqué est noté par R= (Q,M0).

Soit R= (P, T, F, W, M0) un RdP. On a les notations suivantes :

- °t : l'ensemble des places d'entrée de la transition t.
- t° : l'ensemble des places de sortie de la transition t.
- °p : l'ensemble des transitions d'entrée de la place p.
- P° : l'ensemble de transitions de sortie de la place p.

2.2.4 Propriétés des RdP

- **Bornitude :** Un réseau de Petri est considéré comme étant borné si et seulement si toutes ses places le sont. Une place p est dite bornée si, pour tout marquage accessible M, le nombre de jetons dans cette place est inférieur à une constante k. La caractéristique de bornabilité d'un réseau de Petri fournit des indications sur les valeurs bornées des ressources requises par le système. Si un réseau est borné et que cette limite est inférieur ou égale à un, on parle alors d'une bornabilité unitaire.

  Un RdP marqué peut ne pas être borné : dans l'exemple représenté à la Figure 2.1, la transition T1 admet la place P1 comme unique place d'entrée. La place P1 a une marque : la transition T1 est franchissable. Comme P1 est aussi place de sortie de T1, le franchissement de T1 ne change pas le marquage de P1. La transition T1 est donc franchissable en permanence et peut donc être franchie un nombre de fois infini. Chaque franchissement de T1 ajoutant une marque dans la place P2, le marquage de celle-ci peut donc tendre vers l'infini.

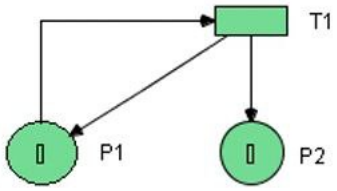


FIGURE 2.1 – RdP non borné

- **Vivacité** L'évolution du marquage d'un réseau de Petri se réalise par le franchissement des transitions. Lorsque certaines transitions ne sont jamais franchies au cours de cette évolution, cela suggère que les événements associés à ces transitions ne se produisent pas et que le marquage partiel du réseau de Petri ne progresse pas. Cela indique un problème potentiel de conception du système, où le sous-système modélisé par cette section pourrait ne pas fonctionner comme prévu. L'objectif est de détecter systématiquement ce phénomène en

analysant les propriétés du modèle de réseau de Petri, afin de fournir un outil d'aide à la conception du système. Une transition t est dite existe par rapport à un marquage initial M0 si, pour tout marquage M accessible à partir de M0, il existe une séquence d'actions partant de M et contenant t. Autrement dit, quel que soit le niveau d'évolution, il est toujours possible d'atteindre la transition t. Un réseau de Petri est dit vivant par rapport au marquage initial M0 si toutes ses transitions sont persistent par rapport à ce marquage initial. Dans un tel réseau, il est assuré que chaque transition peut éventuellement être franchie, quelle que soit l'étape actuelle du système. La propriété de vivacité est une caractéristique puissante mais souvent difficile à vérifier.

Exemples : Les transitions T1 et T2 du RdP marqué Figure 2.2 sont vivantes.

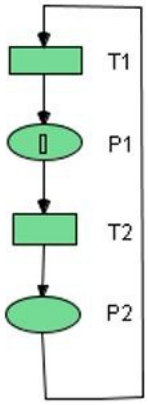


FIGURE 2.2 – RdP vivant

- **Blocage :** Un blocage dans un réseau de Petri correspond à un marquage où aucune transition n'est franchissable. Ainsi, un réseau de Petri est considéré comme étant sans blocage (ou pseudo-vivant) par rapport à un marquage initial M0 si aucun des marquages accessibles Mi ne constitue un blocage.

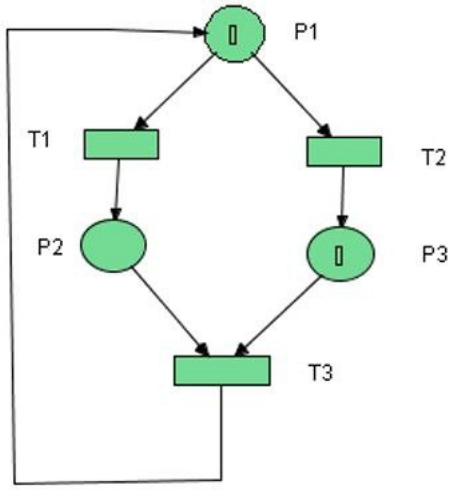


FIGURE 2.3 – Exemple d'un RdP avec blocage

- **Réinitiabilité :** Un réseau de Petri est réinitialisable (ou réversible) : sur l'exemple représente Figure 2.4 :si, pour chaque marquage M, il existe une séquence de transitions qui permet de revenir au marquage initial M0. Cette caractéristique fournit des informations sur les activités répétitives, ce qui est pertinent pour la plupart des systèmes interactifs réels.

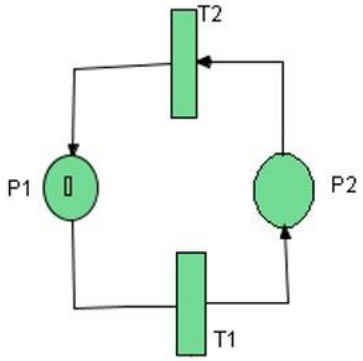


FIGURE 2.4 – Réseau réinitialisable

2.2.5 Avantages du RdP :

Les réseaux de Petri offrent plusieurs avantages :

✓ Ils permettent de décrire la structure d'un système de manière précise tout en restant informels.

- ✓ Ils offrent un support pour la conception graphique, facilitant ainsi la représentation visuelle du système.
- ✓ Ils permettent une approche itérative de la modélisation du système, en le décomposant progressivement en composants plus simples.
- ✓ Ils offrent la possibilité d'intégrer à la fois la structure et la dynamique du système sur un même support.
- ✓ Ils facilitent la transition d'une représentation graphique du système vers une description formelle, favorisant ainsi une analyse mathématique du système pour garantir sa cohérence.

2.2.6 Les méthodes d'analyse des réseaux de Petri

Nous abordons deux méthodes analytiques pour étudier les réseaux de Petri : le graphe de marquages et les équations matricielles. La première méthode consiste à construire un graphe représentant tous les marquages accessibles du réseau, puis à utiliser des techniques de théorie des graphes pour en déduire les propriétés. La deuxième méthode consiste à trouver la représentation matricielle du réseau, puis à utiliser des techniques d'algèbre linéaire pour obtenir ses propriétés. La vérification des propriétés dynamiques se fait généralement en construisant un arbre couvrant, appelé également graphe couvrant. Cet arbre est composé de nœuds correspondant aux marquages accessibles et d'arcs représentant les transitions permettant de passer d'un marquage à un autre. La construction de cet arbre permet de déterminer si le réseau de Petri est limité si le réseau de Petri est limité ou non. Pour un réseau de Petri limité, l'arbre est appelé graphe des marquages accessibles. L'analyse des propriétés structurales repose principalement sur des techniques d'algèbre linéaire. La structure d'un réseau de Petri peut être représentée par une matrice d'incidence, et non par une matrice d'adjacence.

2.2.7 Quelques extensions des réseaux de Petri

Les réseaux de Petri offrent une modélisation efficace pour représenter des comportements cruciaux dans les systèmes tels que la parallélisation, la synchronisation et le partage de ressources. Cependant, cette approche rencontre des limites lorsque la modélisation de systèmes réels conduit à des réseaux de Petri de grande taille, compromettant leur lisibilité et rendant leur manipulation et leur analyse difficiles en raison du problème d'explosion combinatoire. Pour surmonter cette contrainte et étendre leur capacité de modélisation, des extensions des réseaux de Petri, appelées

réseaux de Petri de haut niveau, ont été développées. Ces extensions prennent en compte d'autres aspects tels que la coloration, la gestion du temps, et bien d'autres. Dans la section suivante, nous explorerons certaines de ces extensions des réseaux de Petri.

2.2.7.1 Réseaux de Petri colorés

Les réseaux de Petri colorés sont une extension des réseaux de Petri dans lesquels les jetons sont associés à des couleurs. Chaque couleur représente une information spécifique attachée au jeton. Cette distinction par couleur permet de différencier les jetons et peut être de n'importe quel type. Ainsi, les arcs sont étiquetés non seulement par le nombre de jetons, mais aussi par leur couleur. Le franchissement d'une transition est alors régulé par la présence du nombre requis de jetons dans les places d'entrée, en tenant également compte des couleurs associées aux arcs. Après le franchissement d'une transition, les jetons correspondant aux couleurs des arcs d'entrées ont retirés des places d'entrée, tandis que les jetons correspondant aux couleurs des arcs de sortie sont ajoutés aux places de sortie de la transition pour refléter les changements. Les réseaux de Petri colorés n'apportent pas de capacité descriptive supplémentaire par rapport aux réseaux de Petri classiques, mais ils permettent une représentation plus condensée des informations [48].

2.2.7.2 Réseaux de Petri temporisés

Dans les réseaux de Petri classiques, il n'existe pas de notion de temps pour régir le franchissement des transitions ou la durée de vie des marquages. Cependant, de nombreux systèmes événementiels dépendent du temps pour leur évolution, et la prise en compte du temps est cruciale pour évaluer les performances ou aborder les problèmes de planification dans un système dynamique. Pour répondre à ce besoin, les réseaux de Petri temporisés (RdPT) ont été développés. Le premier travail dans ce domaine a été réalisé par Ramchandani, qui a proposé le modèle de réseau de Petri t-chronométré (t-RdPT). Dans ce modèle, chaque transition du réseau de Petri est associée à un paramètre temporel qui représente la durée pendant laquelle cette transition est active. Un autre modèle a été proposé par Sifakis, appelé le réseau de Petri temporel (p-RdPT). Dans ce modèle, un paramètre temporel est associé à chaque place, représentant la durée minimale pendant laquelle un marquage doit rester dans cette place (temps d'indisponibilité). Sifakis a démontré ultérieurement que son modèle et celui de Ramchandani étaient équivalents [63].

2.2.7.3 Réseaux de Petri synchronisés

Dans le modèle de réseau de Petri que nous avons cité précédemment, le franchissement d'une transition indique que toutes les conditions nécessaires pour réaliser cette transition sont réunies, mais le moment précis du franchissement n'est pas déterminé. En revanche, dans les réseaux de Petri synchronisés, chaque transition est associée à un événement spécifique. Le franchissement de la transition se réalise réellement [67]. Dans un réseau de Petri synchronisé, les transitions ne sont pas nécessairement réalisables en permanence. La confirmation de la transition dépend de la satisfaction des conditions inscrites sur la place. Cela devient possible uniquement lorsque l'événement extérieur associé à la transition se produit : à ce moment-là, la transition est immédiatement activée. Si plusieurs transitions peuvent se produire simultanément en fonction du marquage de leurs places d'entrée, seule la transition dont l'événement associé s'est produit en premier est effectivement franchie" pour plus de précision. Ce mécanisme permet de garantir la synchronisation et la cohérence des actions dans le réseau de Petri synchronisé.

2.3 Le paradigme (réseaux dans les réseaux)

Les réseaux dans les réseaux, ou "Nets Within Nets", représentent une approche avancée des réseaux de Petri qui permet l'intégration de réseaux dans des structures dynamiques. Contrairement aux réseaux classiques où les jetons sont des éléments passifs, dans les Nets Within Nets, les jetons sont considérés comme des entités actives, agissant comme des objets. Dans ce modèle, les places et les transitions peuvent contenir ces objets qui représentent généralement des ressources ou indiquent des états de contrôle. Des structures plus complexes sont modélisées en utilisant des jetons typés dans des réseaux de Petri colorés. Cette approche orientée objet implique la conception logicielle comme une interaction entre objets discrets, intégrant à la fois la structure de données et le comportement, ce qui constitue un paradigme de réseaux internes [70].

En général, il existe deux types de sémantique des jetons : la sémantique de la valeur et la sémantique de la référence. Dans la sémantique de la valeur, les jetons sont considérés comme des représentations directes de réseaux, ce qui permet une imbrication hiérarchique des réseaux, les réseaux ne pouvant être situés qu'en un seul endroit. En revanche, dans la sémantique de la référence, les jetons représentent des références à des réseaux, permettant des structures d'imbrication de réseaux arbitraires, hiérarchisées, acycliques voire cycliques.

Dans les sections suivantes, nous donnerons une brève introduction à l'implémentation de

certains aspects des réseaux dans les réseaux, appelés réseaux de référence.

2.3.1 Les réseaux de référence

Les réseaux de référence fournissent une notation graphique particulièrement adaptée pour décrire et exécuter des processus simultanés et complexes. Comme d'autres types de réseaux de Petri, ils disposent d'un outil de simulation appelé RENEW (Reference Network Workshop) [69]. Ces réseaux étendent les concepts des réseaux de Petri classiques et colorés en introduisant de nouveaux éléments, tels que les instances de réseau, la représentation des réseaux en tant que jetons d'objet, la communication via des canaux synchrones et l'utilisation de différents types d'arcs [76].

Cependant, malgré ces ajouts, les réseaux de référence restent très similaires aux réseaux de Petri colorés tels que définis par Jensen. Voici des définitions de ces extensions :

- **Instances de réseau :** Inspirées du principe d'instanciation d'objets à partir de classes dans les langages de programmation orientée objet, les instances de réseau sont des copies instanciées à partir d'un réseau modèle. Il est important de noter que différentes instances d'un même réseau peuvent avoir des états différents à un moment donné et être totalement indépendantes les unes des autres.
- **Réseau en tant que jeton d'objet :** Les réseaux de référence introduisent la notion de modélisation de réseaux à l'intérieur d'autres réseaux. Les places dans ces réseaux (appelés également réseaux système) peuvent contenir des jetons représentant un autre réseau (appelés réseaux d'objets). Ainsi, une hiérarchie de réseaux peut être facilement établie, où un réseau système contenant un jeton de réseau d'objet peut lui-même être considéré comme un jeton représentant un réseau d'objet dans un autre réseau système.
- **Canaux de synchronisation :** Initialement proposés par Christensen et Hansen, les canaux synchrones permettent de synchroniser deux transitions à exécuter simultanément. Pour que cette synchronisation soit possible, les deux transitions doivent avoir le même nom de canal et le même nombre de paramètres. Il est également possible de synchroniser plus de deux transitions à un instant donné en les reliant à plusieurs canaux de synchronisation.
- **Types d'arcs :** En plus des arcs classiques, les réseaux de référence proposent deux autres types d'arcs : les arcs prédéfinis et les arcs de test. Ces deux types d'arcs partagent la caractéristique de ne pas modifier le marquage associé à une place. L'arc prédéfini, en réalité,

est une façon alternative de représenter deux arcs classiques orientés dans des directions opposées. Il permet de prédéfinir des jetons lors de l'exécution de transitions. Les arcs de test, quant à eux, sont unidirectionnels et permettent l'accès à un jeton spécifique sans modifier le marquage de la place.

2.3.2 Modélisation de la mobilité

Visuellement, tout comme les jetons noirs ou colorés utilisés dans les réseaux de Petri classiques, les réseaux de référence permettent à une place de contenir un jeton représentant un autre réseau de Petri. Cette représentation est illustrée dans les figures 2.5 et 2.6.

Cependant, cette approche peut s'avérer peu pratique lors de la modélisation de systèmes très complexes. Pour résoudre ce problème, l'outil de modélisation RENEW utilise un type de pointeur de référence, d'où le nom "réseau de référence", pour contenir le réseau de jetons. Ainsi, dans la terminologie des réseaux de référence, le réseau de jetons est appelé réseau objet, tandis que le réseau le contenant est appelé réseau système.

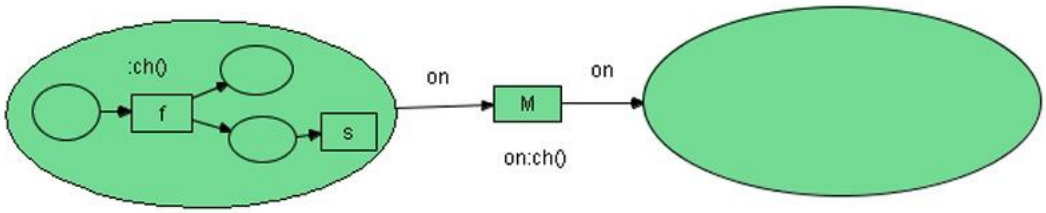


FIGURE 2.5 – Un réseau objet contenu dans un réseau système

Dans le réseau présenté, la place de gauche contient un réseau d'objets avec deux transitions et trois places, où la première place contient un jeton (représenté par "[]") et les deux places restantes sont vides. La transition "f" est associée au canal de synchronisation " : ch()" (considéré comme un alias pour la transition "f" pour simplifier la compréhension). Ce réseau d'objets est nommé "on".

Le réseau système comprend une transition et deux places, où la première place contient le réseau d'objets "on" et la seconde place est vide. Le libellé "on" sur la transition "M" indique le type de jeton à retirer (arc en entrée) et le compteur (arc en sortie). L'exécution de la transition "M" entraîne le retrait synchronisé et le placement correct du réseau d'objets de la place gauche dans la position appropriée. La transition "M" est déclenchée par la méthode "on : ch()", ce qui

signifie que la transition ”f” doit être validée préalablement (la transition ”f” est conditionnée par la méthode ” :ch()” avant que la transition ”M” ne puisse être activée).

Dans la figure 2.5, cette synchronisation entre le réseau d'objets (transition ”f”) et le réseau système (transition ”M”) est possible. Cependant, dans la figure2.6, le réseau d'objets a été déplacé à la position correcte, ce qui rend la transition ”f” non exécutable, car il n'y a plus d'intersection. Ainsi, la transition ne peut plus être déclenchée de manière synchronisée.

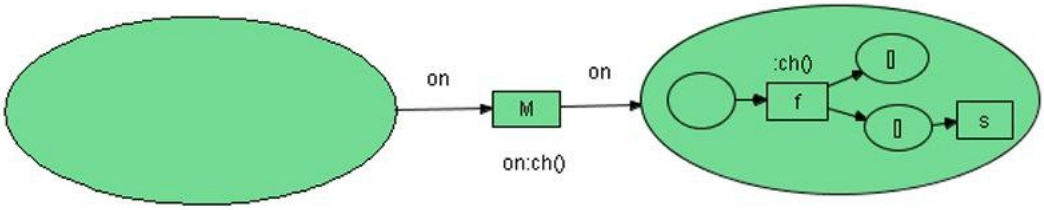


FIGURE 2.6 – Un réseau objet contenu dans un réseau ststème

Cet exemple illustre comment les interactions entre les réseaux d'objets et les réseaux système peuvent être exploités pour modéliser le déplacement d'une entité d'un emplacement à un autre. Le réseau de système peut ainsi offrir ou refuser des possibilités de mouvement, permettant ainsi aux moments appropriés, le déplacement du réseau d'objets en activant les transitions correspondantes.

Les transitions sont déclenchées en fonction de la condition préalable du canal de synchronisation (transitions ”f”), tandis que les transitions modélisent le mouvement dans le réseau de système (transitions ”M”).

Sans cette approche (où les jetons sont considérés comme des réseaux), le modélisateur devrait représenter ou encoder l'entité mobile en utilisant un formalisme différent, ce qui pourrait conduire à l'ignorance des actions internes de ces entités mobiles. Ces actions seraient alors attribuées au modélisateur du réseau de système, ce qui serait en contradiction avec la réalité.

2.3.2.1 Types de mobilité

L'interaction entre le réseau d'objets et le réseau système induit quatre possibilités pour qu'un réseau d'objets se déplace ou soit déplacé [44].

- **Mouvement Spontané :** Se produit sans influence du réseau d'objets ou du réseau système, sans conditions préalables (Figure 2.7).

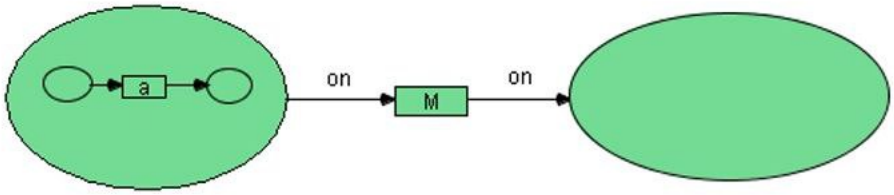


FIGURE 2.7 – Mouvement Spontané

- **Mouvement Subjectif :** Le réseau système définit les mouvements possibles, mais un réseau d'objets peut contrôler certains déplacements dans un environnement donné (Figure 2.8).

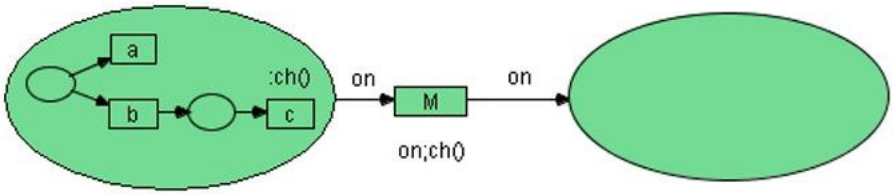


FIGURE 2.8 – Mouvement Subjectif : Le réseau d'objets déclenche le mouvement

- **Mouvement Objectif :** L'ajout d'une condition pour la transition M permet au réseau du système de contrôler le déplacement au réseau d'objets (Figure 2.9).

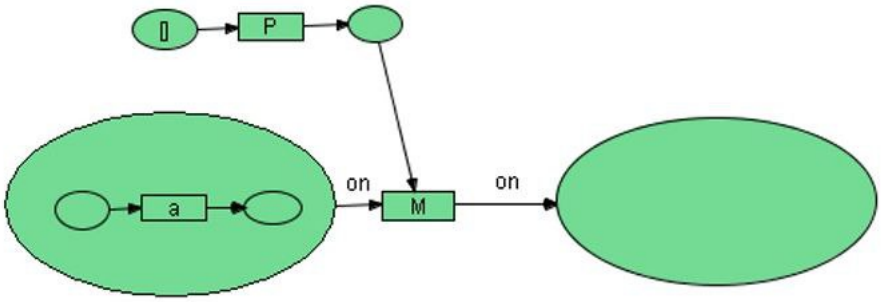


FIGURE 2.9 – Transport : Le réseau du système déclenche le mouvement

- **Mouvement Consensuel :** La transition M nécessite l'accord entre le réseau d'objets et le réseau du système, comme illustré dans la Figure 2.10, où une garde surveille les conditions de mouvement pour autoriser le déplacement.

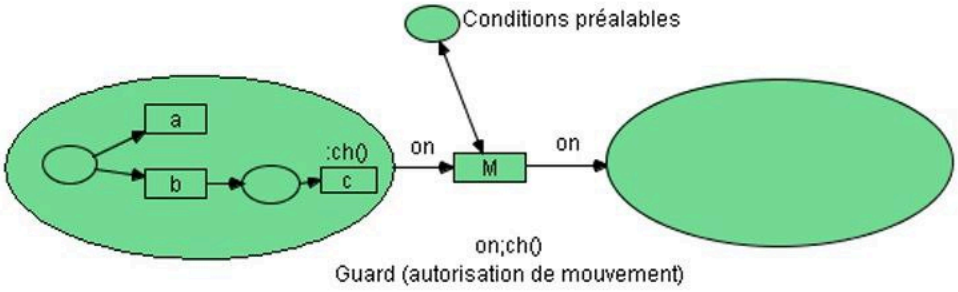


FIGURE 2.10 – Mouvement Consensuel

En utilisant un modèle de réseau dans un réseau, nous pouvons capturer efficacement la concurrence entre les systèmes et les entités mobiles dans un seul modèle, sans sacrifier la précision de l'abstraction.

Le modèle de réseau dans les réseaux, en considérant les jetons comme des réseaux autonomes, offre une approche puissante pour modéliser des systèmes où les entités interagissent de manière complexe et hiérarchique. Cette approche permet de représenter des processus dynamiques et des interactions entre des composants à différents niveaux de granularité. Cependant, pour modéliser des systèmes où les entités possèdent une autonomie et une capacité de décision, une transition vers une architecture multi-agents est nécessaire.

Dans une architecture multi-agent comme MULAN, chaque réseau d'objets dans le modèle de réseau devient un agent autonome. Ces agents sont des entités intelligentes capables de percevoir leur environnement, d'analyser les informations, de prendre des décisions et d'agir en conséquence. Chaque agent agit de manière indépendante pour atteindre ses propres objectifs, tout en tenant compte des actions des autres agents et de l'état global du système.

Les transitions entre les réseaux d'objets dans le modèle réseau sont désormais des interactions entre les agents dans l'architecture multi-agent. Ces interactions peuvent inclure des échanges d'informations, des collaborations pour résoudre des problèmes complexes, des négociations pour parvenir à des accords ou des conflits pour la résolution de ressources limitées. Chaque transition entre les réseaux d'objets peut être interprétée comme une interaction entre les agents correspondants, où ils coordonnent leurs actions pour atteindre des objectifs communs ou pour répondre aux exigences du système.

Ainsi, en passant du modèle de réseau dans les réseaux à une architecture multi-agent comme MULAN, nous obtenons une représentation plus détaillée et significative des systèmes complexes.

Cette transition permet de capturer de manière plus précise les interactions entre les entités autonomes, offrant ainsi une compréhension plus approfondie des dynamiques et des comportements émergents du système.

2.3.3 Multi Agent Architecture MULAN

Dans les systèmes logiciels contemporains [17, 41, 50], les agents et les systèmes multi-agents (SMA) jouent un rôle central en tant que concepts structurants. Ils offrent un cadre sophistiqué, dépassant le développement logiciel orienté objet traditionnel, en incorporant des attributs tels que l'autonomie, la coopération, l'adaptabilité et la mobilité. Cela correspond étroitement aux principes de l'architecture MULAN, qui met l'accent sur la conception modulaire, l'adaptabilité et le traitement distribué. Les agents au sein des systèmes MULAN présentent des comportements qui vont au-delà des objets statiques, permettant une interaction dynamique et une collaboration entre entités autonomes. L'architecture MULAN est utilisée pour décrire la hiérarchie des aspects d'un système multi-agent. MULAN a une structure générale comme le montre la Figure 3.1. L'architecture du système multi-agent Mulan est basée sur le paradigme des "réseaux à l'intérieur des réseaux", qui est utilisé pour décrire les hiérarchies naturelles dans un système d'agents. Mulan est implémenté dans Renew. Mulan présente une structure générale comme illustré dans la figure 2.11. Chaque boîte décrit un niveau d'abstraction en termes de réseau système. Chaque réseau système contient des réseaux d'objets, dont la structure est rendue visible par les lignes de ZOOM.

Le réseau dans le coin supérieur gauche décrit un système d'agents, dont les places contiennent des plateformes d'agents en tant que jetons. Les transitions décrivent des canaux de communication ou de mobilité, qui constituent l'infrastructure. Il s'agit simplement d'un exemple illustratif, Le nombre de places et de transitions, ainsi que leur interconnexion, n'ont pas de signification supplémentaire.

En zoomant sur le jeton de plateforme dans la place p1, la structure d'une plateforme devient visible, comme illustré dans la boîte en haut à droite. La place centrale "agents" héberge tous les agents qui se trouvent actuellement sur cette plateforme. Chaque plateforme offre des services aux agents, certains d'entre eux étant indiqués dans la figure. Les agents peuvent être créés (transition "new") ou détruits (transition "destroy"). Les agents peuvent communiquer par échange de messages. Deux agents sur la même plateforme peuvent communiquer via la transition "communication interne", qui relie deux agents, l'émetteur et le récepteur, pour faire passer un message sur un canal synchrone. La "communication externe" ne lie qu'un seul agent, car l'autre agent

est lié à une deuxième plateforme quelque part ailleurs dans le système d'agents. Des facilités de mobilité sont également fournies sur une plateforme : les agents peuvent quitter la plateforme via la transition "envoyer agent" ou les transitions peuvent entrer dans la plateforme via la transition "recevoir agent" en provenance d'une autre plateforme.

Les agents sont également modélisés en termes de réseaux. Ils sont encapsulés, car le seul moyen d'interaction est le passage de messages. Les agents peuvent être intelligents, car ils ont accès à une base de connaissances. Le comportement de chaque agent est décrit en termes de protocoles, qui sont eux-mêmes des réseaux. Les protocoles sont représentés en tant que modèles sur la place "protocoles". Les modèles de protocoles peuvent être instanciés, par exemple lorsqu'un message arrive. Un protocole instancié fait partie d'une conversation et se trouve dans la place "conversations". La structure détaillée des protocoles et leur interaction ont été abordées précédemment dans [44] ; nous éviterons donc les détails ici.

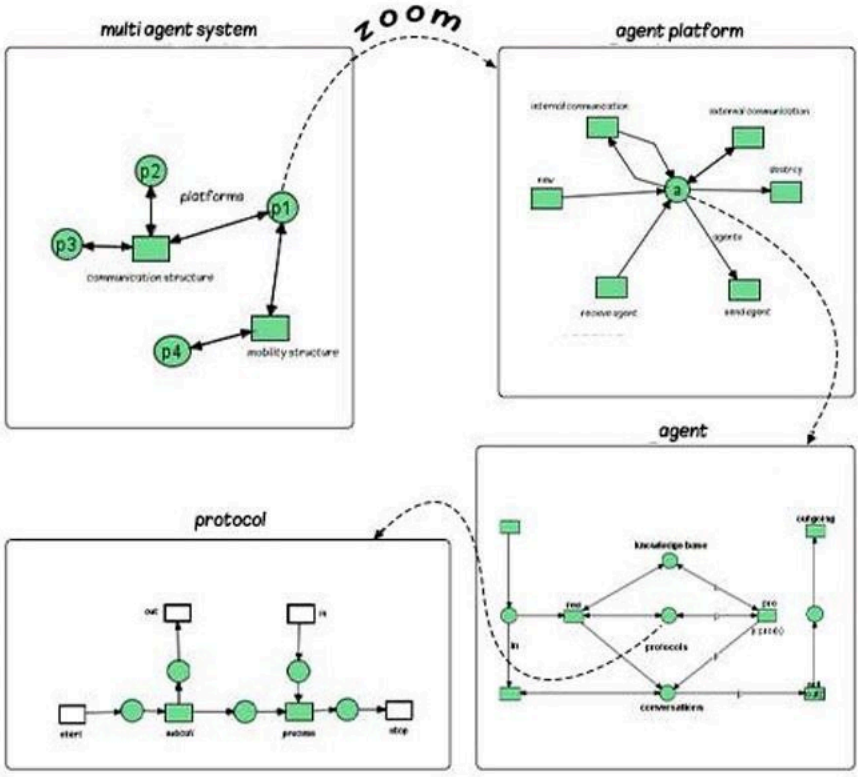


FIGURE 2.11 – Système multi-agents modélisé avec les réseaux dans les réseaux [57]

2.4 Outils de modélisation des réseaux de Petri

Les aspects formels des réseaux de Petri offrent aux développeurs une diversité d'outils pour simuler et vérifier ces réseaux par rapport aux méthodes de vérification de modèle. Parmi ces outils figurent CPNTools, CPNAMI, PROD, JARP, MARIA, LOLA, Petri Net Kernel, GreatSPN, Artifex, ExSpect, FLOWer, f-net, GD ToolKit, Helena, HPSim, JFern, JPetriNet, Opera, ORIS, PACE, RENEW, et al. La plupart de ces logiciels proposent une interface graphique permettant d'éditer les réseaux de Petri et offrent la possibilité de simuler des modèles ainsi que d'analyser les propriétés générales de ces réseaux.

2.5 Conclusion

La transition vers l'intégration des réseaux de Petri, notamment à travers le concept de réseaux dans les réseaux, combinés à l'utilisation des réseaux de référence et de l'architecture MULAN, présente un avantage crucial pour assurer une sensibilité au contexte dans les systèmes multi-agents mobiles. En adoptant une approche plus intégrée et réaliste de la modélisation des systèmes, ces méthodes permettent de prendre en compte efficacement les variations et les spécificités du contexte dans lequel opèrent les systèmes.

L'avantage principal réside dans la capacité à capturer et à traiter les informations contextuelles pertinentes, telles que la localisation des agents, les conditions environnementales, les ressources disponibles, ou encore les exigences spécifiques des tâches à accomplir. En intégrant ces informations dans la conception des systèmes, il devient possible d'adapter dynamiquement les stratégies de fonctionnement en fonction du contexte actuel, ce qui améliore considérablement les performances, l'efficacité et la qualité des services rendus.

Dans le chapitre suivant, nous aborderons les réseaux véhiculaires ad hoc (VANET) en mettant en avant leur rôle crucial dans les environnements intelligents et connectés.

Chapitre 3

Les réseaux véhiculaires (VANET)

3.1 Introduction

Dans les chapitres précédents, nous avons discuté de la sensibilité au contexte et des réseaux dans les réseaux, qui représentent des réseaux de Petri de haut niveau. Dans ce chapitre, nous nous concentrons sur les réseaux véhiculaires ad hoc (VANET) [7, 23, 58], une technologie essentielle pour les véhicules connectés. Nous présenterons leur architecture, leurs modes de communication, ainsi que les services et applications qu'ils offrent. Enfin, nous aborderons les protocoles de routage et l'intégration des approches basées sur les systèmes multi-agents (SMA) dans les réseaux véhiculaires.

3.2 Architecture du réseau VANET

Dans un réseau de véhicules ad hoc, les véhicules sont considérés comme des nœuds pouvant se déplacer librement dans l'environnement et rester connectés au réseau à tout moment. La communication sur le réseau VANET est établie via un dispositif appelé DSRC (Dedicated Short Range Communications) spécialement conçu pour les communications à courte portée [90].

Un réseau VANET se compose de nombreux composants interagissant pour effectuer diverses tâches [5]. En effet, selon les types de communication introduits ci-dessus, nous distinguons trois domaines de communication.

- **Domaine du véhicule :** il s'agit de la communication et de la coordination de plusieurs unités d'application (UA) au sein du même véhicule. Les unités sont chargées d'effectuer une ou

plusieurs tâches en coordination avec d'autres systèmes dans le même véhicule. Un système d'application est constitué d'un ensemble d'éléments interconnectés, tels que les unités de commande, les capteurs et les actionneurs, qui permettent de mettre en œuvre des fonctions spécifiques du véhicule.

- **Domaine Ad-Hoc :** Dans ce domaine, chaque véhicule est équipé d'une unité de communication (On-Board Unit - OBU) qui lui permet de communiquer avec les systèmes installés dans d'autres véhicules. Cela permet la coopération entre les véhicules via la communication V2V (vehicle-to-vehicle), facilitant l'exécution de tâches communes et l'atteinte d'objectifs communs [77].
- **Domaine d'infrastructure :** Le réseau VANET se compose de plusieurs nœuds, tels que les véhicules, les unités de communication d'infrastructure (RSU) et les points d'accès (HS), accessibles par différents usagers de la route via la communication V2I (vehicle-to-infrastructure). De plus, plusieurs systèmes embarqués dans différents véhicules peuvent également communiquer entre eux et effectuer des tâches spécifiques [42, 28].

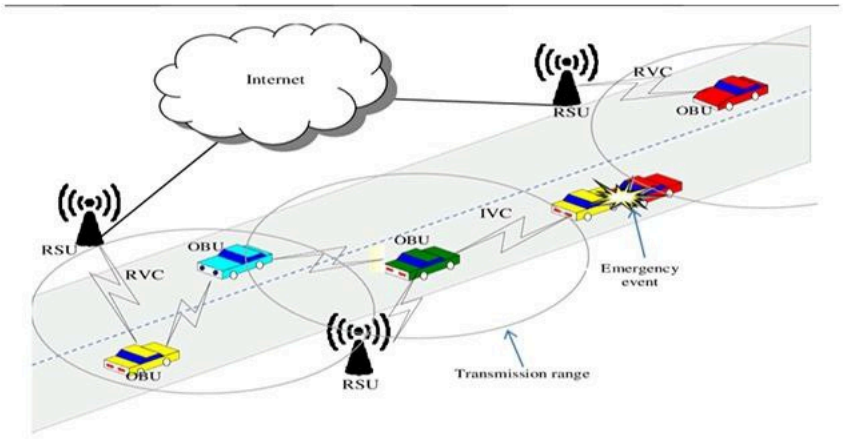


FIGURE 3.1 – Les entités de communication

3.2.1 Modes de communication

Dans les réseaux VANETs, on trouve principalement, des entités fixes qui constituent l'infrastructure (RSU et TA) et les entités mobiles (les véhicules). Pour échanger les informations et données liées à la sécurité et au confort des usagers de la route, ces différentes entités doivent éta-

blir des communications entre elles. Pour cette raison, on distingue trois types de communications véhicule à véhicule (V2V) [14, 2] et véhicule à infrastructure (V2I) [21, 29] et hybride.

3.2.1.1 Communication en mode Ad hoc (V2V)

Dans ce modèle, les réseaux de véhicules sont considérés comme un cas particulier des réseaux mobiles auto-organisés (MANET), où les contraintes d'énergie, de mémoire et de puissance de calcul sont assouplies, et le modèle de mobilité n'est pas stochastique mais prévisible, avec une mobilité très élevé. Cette architecture peut être utilisée dans des scénarios de diffusion d'alertes (freinage d'urgence, collision, décélération, etc.) ou de conduite coopérative. En effet, dans le contexte des applications de sécurité routière, les réseaux d'infrastructures montrent leurs limites, notamment en matière de latence. De toute évidence, la communication ad hoc multi-sauts est plus efficace que la communication via les réseaux ou infrastructure des opérateurs. La figure 3.2 illustre ce mode de communication entre véhicules [64].

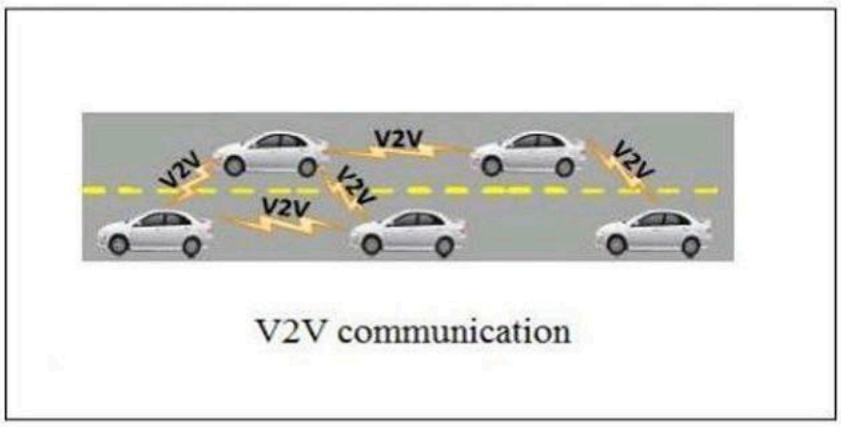


FIGURE 3.2 – mode de communication v2v [64].

3.2.1.2 Communication en mode infrastructure (V2I)

Ce modèle de communication utilise l'infrastructure routière (cellules routières) pour fournir des services tels que l'accès à Internet, la communication entre les véhicules et le personnel de maintenance pour les diagnostics à distance, etc. Ce mode n'est pas adapté aux applications liées à la sécurité routière car le réseau d'infrastructure n'est pas efficace en termes de temps de réponse [54].

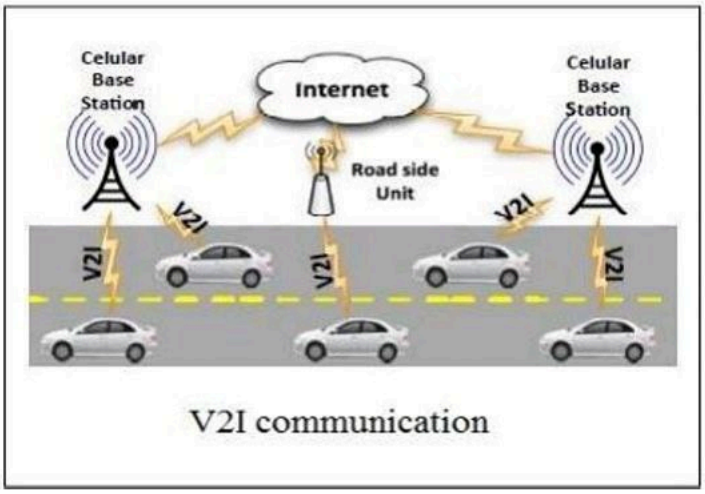


FIGURE 3.3 – Modèles de communication véhicule-infrastructure [53].

3.2.1.3 Communication en mode hybride

Ce type de communication est une combinaison des deux modes de communications précédents V2V si le véhicule destinataire se trouve dans la portée de transmission ou bien en mode V2I si la destination se trouve hors de cette portée de transmission. Ces deux modes sont généralement combinés pour réaliser une communication hybride très intéressante [53].

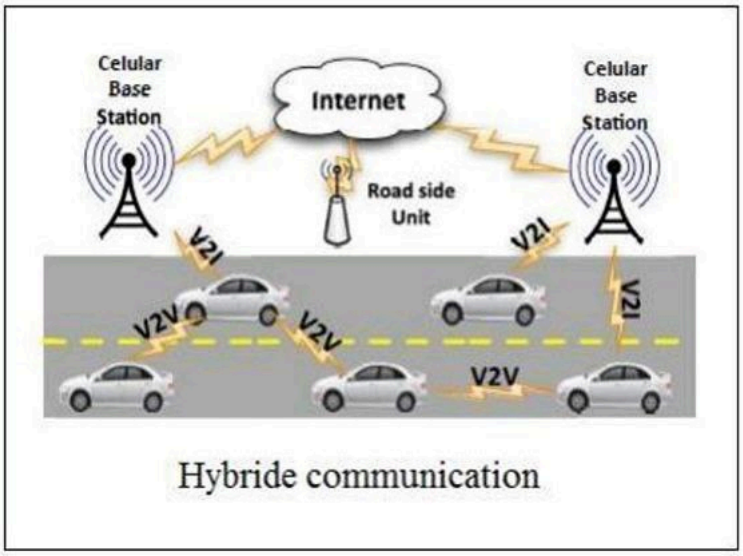


FIGURE 3.4 – Mode de communication hybride [53].

3.2.2 Types de messages

3.2.2.1 Messages de contrôle

Ce type de message contient souvent des informations sur l'identité et l'état actuel du véhicule (position, vitesse, direction, etc.) et sert principalement à identifier les environs. Ce type de message joue un rôle important dans la plupart des protocoles de routage et de sécurité.

3.2.2.2 Messages d'alerte

Ce type de message est envoyé pour avertir les autres véhicules des diverses urgences et catastrophes sur la route (accidents, embouteillages, informations météorologiques, passage de véhicules d'urgence, etc.) afin qu'ils puissent agir. Ce type de message contribue à améliorer la circulation et la sécurité routière.

3.2.3 Les caractéristiques des VANETS

Certaines caractéristiques des VANET sont similaires à celles des MANET, mais il existe également des caractéristiques propres aux VANET qui peuvent être classées comme suit :

a) Topologie hautement dynamique Des vitesses élevées des véhicules et la disponibilité de plusieurs options d'itinéraire caractérisent la topologie dynamique du VANET.

b) Mobilité de la modélisation et de la prévention Prédire la position et le mouvement des véhicules est très difficile. De plus, la modélisation et les prévisions de mobilité dans VANET sont basées sur la disponibilité de modèles de route prédéfinis. La vitesse des véhicules est considérée comme une mesure importante pour une conception efficace du réseau.

c) Environnement de communication Le modèle de mobilité peut avoir différentes caractéristiques selon l'architecture routière, les autoroutes ou les environnements urbains. À cet effet, la communication dans ces situations doit être traitée avec soin.

d) Contrainte de délai difficile En cas d'urgence, la livraison de messages urgents devient un problème critique. Il est donc plus important de faire face à de telles situations que de se contenter de débits de données élevés

e) Interaction avec les capteurs embarqués Les capteurs peuvent lire les données relatives à la vitesse du véhicule, direction et peuvent communiquer avec le centre de données. Ainsi,

les capteurs peuvent être également utilisés pour la formation de liens et dans les protocoles de routage [80].

f) Batterie et stockage illimités Les nœuds dans les VANET ne souffrent pas d'alimentation et de limitation de stockage comme dans les réseaux de capteurs, ainsi le cycle de service n'est pas aussi pertinent que dans les réseaux de capteurs.

3.2.4 Les services offerts par les réseaux VANET

a. Service de sécurité routière Ces services sont des applications ayant un impact direct sur la sécurité des personnes et des biens, c'est-à-dire des applications permettant de réduire le nombre d'accidents routiers et d'améliorer les conditions de circulation [39].

b. Les services liés au confort Le réseau VANET fournit non seulement des services liés à la sécurité des véhicules et de leurs passagers, mais assure également le confort lors des déplacements [64]. Ces services incluent la messagerie instantanée, les jeux en réseau, l'accès à Internet, les paiements automatisés et la diffusion d'informations utiles sur la disponibilité du stationnement, en indiquant aux conducteurs les places disponibles. La gamme de ces services est actuellement très large et offre des perspectives intéressantes aux transporteurs, car ces services ont le potentiel de générer des bénéfices supplémentaires.

3.2.5 Les applications des VANETS

VANET propose centaines d'applications recommandées. Ces applications peuvent être divisées en deux catégories : les applications liées à la sécurité et les applications non sécurisées (voir [1, 33, 6]) : De plus, elles peuvent être classés en applications OBU à OBU, OBU à RSU. Voici une liste de certaines de ces applications.

- **Avertissement coopératif de collision :** L'avertissement coopératif de collision est une application de sécurité entre les appareils embarqués. S'il y a un changement soudain de vitesse ou de direction, le véhicule sera considéré comme anormal et enverra une alerte aux véhicules suivants pour les avertir du danger. Cette application nécessite un algorithme de diffusion efficace avec une latence très faible.

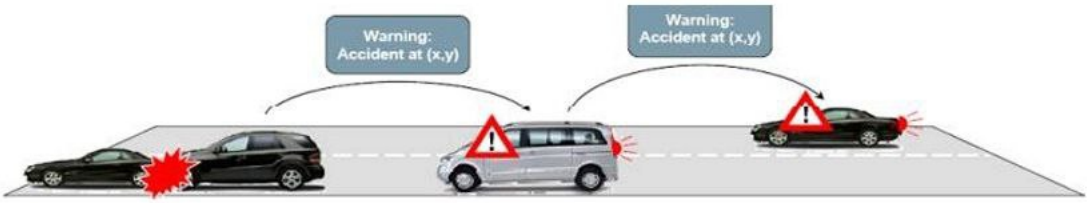


FIGURE 3.5 – réduire les accidents

- **Actions des véhicules d'urgence :** L'approche de véhicules de secours est une application entre les unités embarquées (OBU à OBU) pour la sécurité publique. Les véhicules d'urgence rapides (ambulances et voitures de police) peuvent avertir les autres véhicules de quitter leur voie. Cette application s'appuie également sur la diffusion efficace.

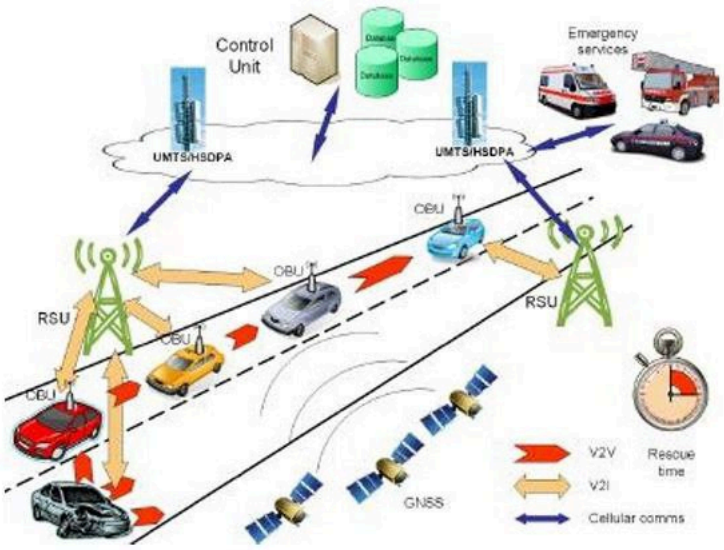


FIGURE 3.6 – rapport d'accident

- **Communication entre véhicules :** La communication de véhicule à véhicule est une application de communication non sécurisée de véhicule à équipement qui permet aux voyageurs de communiquer entre eux via le transfert instantané de fichiers, la vidéoconférence et même le chat vidéo.
- **Paiement de parking :** C'est une application OBU-à-RSU non sécurisée qui offre des avantages aux exploitants de parkings, en simplifiant les paiements des clients et en réduisant les embouteillages aux entrées et sorties de stationnement

- **Gestion du trafic :** La navigation automobile est une application non sécurisée conçue pour réduire le temps de trajet et la consommation de carburant en échangeant des informations en temps réel sur les conditions de circulation sur les voies.
- **Avertissement de collision aux intersections :** Une application de sécurité de l'OBU à RSU. Aux intersections, un nœud central avertit les véhicules en approche d'un accident potentiel et les aide à déterminer la vitesse appropriée. Cette application utilise uniquement des messages diffusés.

3.2.6 Classification des protocoles de routage dans les réseaux VANET

La principale caractéristique des réseaux de véhicules est la grande mobilité, qui permet des topologies très dynamiques. Cette caractéristique signifie que les protocoles de routage MANETS traditionnels ne conviennent pas aux VANET. En fait, dans certains environnements de communication tels que les autoroutes, la vitesse de VANET peut être bien plus rapide que celle de MANETS. Diverses solutions ont été proposées pour le routage dans les réseaux VANET. Faites la distinction entre deux classes de protocoles : les protocoles basés sur la topologie[A .Yacinak], qui sont classés en protocoles proactifs, réactifs et hybrides qui créent des routes, et les protocoles basés sur l'emplacement (géographique) qui utilisent l'emplacement physique du nœud mobile pour configurer le routage.

a. Le routage basé sur la topologie

i. Les protocoles proactifs

Le principe de base des protocoles proactifs est de précalculer les routes et de maintenir la table de routage à jour en permanence, afin que lorsqu'un nœud souhaite envoyer un paquet à un autre nœud, la route soit immédiatement connue. Étant donné que les nœuds d'un réseau mobile ad hoc peuvent apparaître et disparaître de manière aléatoire, un système doit être mis en place pour échanger en permanence des paquets de contrôle. Grâce à ces paquets, la table de routage est maintenue. Cette approche permet aux nœuds de construire la topologie du réseau de manière distribuée. Lorsqu'un nœud reçoit un paquet de contrôle, il met à jour sa table de routage. Par conséquent, de nouvelles routes sont créées sur la base des informations topologiques envoyées par les trames de contrôle. Ce processus est déclenché chaque fois que la topologie change

pour reconstruire les routes. L'un des avantages de ces protocoles est que les itinéraires sont facilement disponibles en cas de besoin. Cependant, le trafic généré par l'échange de paquets de contrôle réduit la bande passante.

- **Le protocole FSR (Fisheye State Routing)**

  Chaque nœud maintient une table topologique basée sur les dernières informations reçues des nœuds voisins. Pour réduire la taille des messages de commande dans les grands réseaux, des périodes d'échange différentes sont utilisées pour chaque entrée de la table de routage. L'inconvénient de FSR est qu'à mesure que la taille du réseau augmente, la table de routage s'agrandit et la découverte de route peut échouer si le nœud de destination est en dehors de la portée du nœud source.

- **Le protocole OLSR (Optimized Link State Routing Protocol)**

  Il s'agit d'un protocole de routage destiné aux réseaux maillés, sans fil ou mobiles. Un concept clé utilisé dans ce protocole est l'utilisation du relais multipoint (MPR). L'ensemble MPR est choisi pour couvrir tous les nœuds à deux sauts. Agissant comme un protocole proactif, les informations topologiques sont régulièrement échangées avec d'autres nœuds du réseau.

ii. Les protocoles réactifs

Le principe des protocoles réactifs, également appelés protocoles de routage à la demande, est de lancer le processus de recherche d'itinéraire uniquement lorsque cela est nécessaire. Cela permet d'économiser de la bande passante et de l'énergie. Lorsqu'un paquet doit être envoyé d'un nœud source vers un nœud de destination, les protocoles de routage trouvent le chemin vers la destination. Une fois ce chemin trouvé, il est entré dans la table de routage et peut être utilisé s'il est accessible ou jusqu'à ce que la route devienne inutilisable. Cette découverte se fait généralement par inondation, c'est-à-dire la transmission progressive de paquets de recherche de chemin sur l'ensemble du réseau ou sur des parties du réseau. Dans ce cas, la bande passante sera plus élevée. Cependant, la création d'un itinéraire prend plus de temps que les protocoles proactifs, car l'itinéraire vers la destination n'est pas immédiatement disponible.

- **Le protocole AODV (Ad hoc On Demand Distance Vector)**

  Un protocole de routage pour les réseaux ad hoc. Convient au routage unicast et multicast. Il est sans boucle, démarre automatiquement et peut accueillir un grand nombre de nœuds mobiles. Lorsqu'un nœud source demande des routes, celles-ci

sont créées à la volée et ces routes sont conservées aussi longtemps que la source en a besoin. Ce protocole de routage est économe en énergie et ne nécessite pas beaucoup de puissance de calcul, ce qui le rend facile à installer sur de petits appareils mobiles.

- **Le protocole DSR (Dynamic Source Routing)**

  Il s'agit d'un protocole de routage pour les réseaux maillés sans fil. Il est similaire à AODV dans la mesure où il forme des itinéraires à la demande lorsque l'ordinateur souhaite les envoyer. Cependant, au lieu de s'appuyer sur la table de routage de chaque routeur intermédiaire, il utilise le routage source.

iii. Les protocoles hybrides

Les protocoles hybrides combinent les deux approches précédentes. Ils utilisent des techniques proactives dans un petit rayon autour de la source dans lequel le nombre de sauts est très faible (par exemple 3-4 sauts), et des techniques réactives sur des nœuds plus distants. L'avantage des protocoles hybrides est qu'ils sont plus adaptables aux grands réseaux. Cependant, ce type de protocole combine les inconvénients des protocoles proactifs et réactifs, notamment. Ils échangent des paquets de contrôle périodiques et inondent l'ensemble du réseau pour trouver des routes vers des nœuds distants.

- **Le protocole ZRP (Zone Routing Protocol)**

  Il s'agit d'un modèle hybride de méthodes proactives et réactives. Le principal problème du développement de protocoles de routage pour les réseaux ad hoc est de déterminer l'itinéraire d'un paquet de données, le nœud source doit connaître les informations qui lui permettent d'atteindre au moins le nœud le plus proche. En revanche, La topologie de ces réseaux change fréquemment. De plus, avec l'augmentation du nombre de nœuds et donc de cibles potentielles, l'échange de données devient plus fréquent et à grande échelle. Par conséquent, la quantité de données de mise à jour du trafic peut être énorme. Ce qui est contradictoire avec le fait que toutes les mises à jour dans un réseau connecté ad hoc sont effectuées par voie hertzienne et sont donc gourmandes en ressources.

  Il divise l'ensemble de réseau en zones qui ne se chevauchent pas. L'objectif est d'établir un itinéraire stable de la source à la destination afin de réduire les retards. Il permet de rechercher des itinéraires entre zones afin de limiter les inondations au sein du réseau et de sélectionner le meilleur itinéraire en fonction de critères de

stabilité.

HARP est mis en œuvre à deux niveaux : intra-zone et inter-zone, Non applicable aux réseaux ad hoc à haute mobilité.

- **Le protocole Gy-TAR (Greedy Traffic-Aware Routing Protocol)**

  Le protocole de routage Gy-TAR est un protocole géographique basé sur la localisation, adapté aux réseaux de véhicules et capable de trouver des chemins robustes dans les environnements urbains. Chaque véhicule connaît son emplacement grâce à une unité Global Positioning System (GPS), l'émetteur connaît l'emplacement du récepteur grâce à des services de localisation tels que GLS (Grid Location Service), et en un seul saut, vous pouvez déterminer l'emplacement des nœuds adjacents. Grâce à des cartes numériques

- **Le protocole HCB (Hard Core Bitch)**

  Il s'agit d'un protocole de routage hiérarchique basé sur des clusters, conçu pour les réseaux ad hoc hautement mobiles. Son architecture de communication repose sur deux couches. Au niveau de la couche 1, la plupart des nœuds disposent d'une seule interface radio et utilisent le multisaut pour communiquer entre eux. Certains de ces nœuds disposent également d'interfaces distinctes pour la communication sans fil longue portée, appelées super-nœuds, aux couches 1 et 2. Les supernodes peuvent communiquer entre eux via des stations de base de couche 2.

b. Le routage basé sur la géographie

Les protocoles de routage géographique (ou basés sur la position) utilisent des coordonnées géographiques pour déterminer le chemin vers une destination. . Chaque nœud dans le réseau inclut l'identifiant et la position de la destination dans le paquet à envoyer. Ces protocoles réduisent la signalisation, notamment dans les réseaux larges et dynamiques [a].

Les protocoles de routage géographique sont classés en trois sous-classes :

- Basés sur la position géographique
- Basés sur la cartographie routière
- Basés sur le trafic routier

Quelques exemples de protocoles notables sont :

- **GSR (Geographic Source Routing) :** Combine le routage basé sur la localisation avec la topologie des routes pour optimiser le routage en milieu urbain [22].

- **A-STAR (Anchor-based Street and Traffic Aware Routing) :** Utilise les itinéraires d'autobus pour identifier des routes d'ancre dans les environnements urbains, tenant compte du trafic pour calculer les chemins optimaux [59].
- **VADD (Vehicle-Assisted Data Delivery) :** Exploite le mouvement prévisible des véhicules et les informations sur le trafic routier pour estimer les délais et optimiser la retransmission des paquets [59].

3.2.7 Exploration des Approches SMA dans les Réseaux Véhiculaires :

Dans cette section, nous approfondirons les approches les plus significatives en matière de réseaux de véhicules intelligents (VANET) qui tirent parti des systèmes multi-agents. Dans [88], les auteurs introduisent un système basé sur l'utilisation d'agents mobiles dans les réseaux de véhicules. Il représente les avantages que peuvent offrir les agents mobiles dans ces réseaux ainsi que les difficultés rencontrées. Dans [20], un système de transport intelligent basé sur le paradigme multi-agents appelé CSCW (Intelligent Computer-Supporté Coopérative Work) aidera les conducteurs de véhicules autour de la zone d'urgence à prendre les décisions appropriées. Dans [21], les auteurs proposent des agents mobiles pour améliorer la capacité des systèmes de gestion du trafic à faire face à l'incertitude dans un environnement dynamique. En particulier, ils développent un système d'agent mobile conforme à la norme IEEE FIPA appelé Mobile-C, et conçoivent un agent basé sur la détection du trafic en temps réel et la gestion du système un (A BRTTDMS) [21]. Dans [12], les auteurs proposent un mécanisme auto-adaptatif de régulation du trafic, qui se concentre sur le comportement aux intersections, en appliquant un agent pour représenter les éléments de l'infrastructure, qui collaborent entre eux pour réduire les embouteillages. Ainsi, ils montrent une solution basée sur des mécanismes qui : (i) impliquent un contrôle dynamique des feux de circulation, et (ii) ne dépendent pas des interactions avec les véhicules .

Dans un autre travail [74], Sadiqa Jafari et al. ont développé une nouvelle méthode de contrôle fondée sur plusieurs intersections en utilisant un contrôle de logique floue prenant en compte le type de feu de circulation et la durée de la phase verte pour réguler l'équilibre idéal aux intersections.

Dans [88], les auteurs proposent l'utilisation de la technologie des agents mobiles pour un traitement efficace des requêtes distribuées dans les VANET. Ils présentent un modèle formel de l'approche proposée en utilisant le formalisme des réseaux de Petri. Krajzewicz et al. [47] discutent des feux de circulation sensibles au flux. Dans leur approche, la taille des files d'attente entrantes à chaque intersection est comparée, puis les véhicules dans la plus grande file d'attente ont la

priorité pour traverser l'intersection. Cependant, cette étude se concentre uniquement sur une seule intersection. Rajendran Sathiyaraj et Ayyasamy Bharathi [73] proposent un système efficace de contrôle et de déviation intelligents des feux de circulation (EITLCD) afin de minimiser le problème de congestion du trafic. Ce système réduit la taille de la file d'attente et fournit un itinéraire alternatif aux véhicules pour éviter les zones congestionnées et fluidifier la circulation.

Dans [84], une approche hybride est proposée, combinant la modélisation Timed Petri Nets et la modélisation modulaire. L'objectif principal de cette combinaison est de garantir le meilleur contrôle de surveillance des flux de trafic urbain aux carrefours à deux phases dans les villes moyennes. Chaque intersection est considérée comme un système multi-agents composé de tous les agents représentant les différents espaces réels. Le modèle de réseau de Petri temporisé, qui régule les cycles des feux de signalisation selon un schéma de timing idéal déterminé par un algorithme génétique continu, supervise les actions des agents connectés.

3.3 Conclusion

Les réseaux véhiculaires ad hoc (VANET) représentent une infrastructure clé pour les systèmes de transport intelligents, en permettant des communications efficaces et des services variés, allant de la sécurité routière au confort des usagers. Ce chapitre a permis de présenter leur architecture, leurs modes de communication et leurs applications. Dans le chapitre suivant, nous explorerons notre contribution qui exploite l'architecture MULAN et le concept de sensibilité au contexte pour proposer une approche dans la conception des systèmes multi-agents mobiles.

Chapitre 4

Modélisation par agents mobiles pour atténuer la congestion dans les réseaux véhiculaires

4.1 Introduction

Le chapitre précédent a présenté les réseaux véhiculaires ad hoc (VANET), en mettant en évidence leurs caractéristiques et les défis associés à leur gestion dans le cadre des systèmes de transport intelligents. Cette analyse a servi à poser les bases de notre approche pour résoudre les problèmes de congestion dans ces réseaux.

Dans cette première contribution, nous abordons la gestion du trafic aux intersections des VANET, en nous appuyant sur les systèmes multi-agents (SMA). Nous explorons comment l'outil Renew, grâce à sa flexibilité et sa capacité de modélisation des interactions, peut être utilisé pour optimiser la gestion du trafic en temps réel, en répondant aux défis complexes de congestion dans ces zones critiques.

Dans ce chapitre, nous proposons une vue d'ensemble de notre approche, suivie d'une analyse détaillée de l'architecture proposée. Nous identifions également les principaux types d'agents impliqués dans la gestion de la congestion, mettant en lumière leur rôle dans la résolution de ce problème aux intersections des réseaux véhiculaires.

4.1.1 Motivation

Les systèmes multi-agents et en particulier les agents mobiles [55, 85] constituent un moyen noble de gérer les données distribuées dans VANET car ils capturent des informations contextuelles

et résolvent les problèmes de contrôle du trafic. Ces systèmes ont la capacité d'être très appropriés dans des environnements mobiles, intelligents et autonomes.

Ainsi, il existe une multitude de raisons qui justifient le choix d'utiliser les agents mobiles

- Réduire la charge des réseaux,
- Surmonter le temps de latence sur le réseau,
- S'exécuter de manière asynchrone et autonome,
- S'adapter dynamiquement aux environnements d'exécution,
- Hétérogènes,
- Robustes et tolérants

Il convient de noter que la modélisation et la simulation sont deux étapes importantes dans le développement de tout système, en particulier lorsqu'un modèle formel prenant en charge des fonctionnalités telles que la mobilité est utilisé.

4.1.2 Objectifs de l'approche

L'objectif principal de notre approche est de développer des solutions robustes pour les systèmes multi-agents afin de relever le défi complexe de la congestion du trafic dans l'environnement des réseaux véhiculaires (VANET). Dans cette optique, des agents mobiles et intelligents sont déployés pour la gestion dynamique des données au sein des VANET, offrant ainsi des solutions agiles face à des situations inattendues. Cette approche repose sur la coopération entre les agents pour choisir la meilleure alternative possible et prendre rapidement des mesures appropriées.

Notre solution se positionne comme une réponse efficace à l'atténuation des embouteillages en se basant sur les communications véhicule à véhicule (V2V) [14, 2] et véhicule à infrastructure (V2I) [21, 29]. En privilégiant ces communications, nous visons à réduire le temps d'attente des conducteurs et à améliorer la fluidité du trafic de manière significative.

L'approche que nous présentons repose sur le paradigme "réseau dans le réseau", détaillé dans les chapitres précédents. Plus spécifiquement, notre attention se concentre sur l'optimisation des phases de régulation des feux de circulation pour assurer un contrôle efficace des intersections signalées. Cette optimisation vise à réduire le temps d'attente des véhicules, en mettant particulièrement l'accent sur les véhicules prioritaires tels que les véhicules de police, de pompiers et d'ambulances.

4.2 Description générale de l'approche proposée

Cette approche multi-agent se focalise sur la gestion dynamique des intersections routières en intégrant de manière cruciale la sensibilité au contexte. Trois catégories d'agents occupent des rôles spécifiques au sein du système : les agents d'intersection, les agents de véhicules et les agents de piétons.

- **Agents de l'intersection :** Ces agents sont spécifiquement dédiés à la supervision et à la coordination des opérations à l'intersection. Ils ajustent les feux de signalisation en temps réel en fonction des conditions du trafic. La sensibilité au contexte permet une adaptation dynamique, prenant en compte des facteurs tels que la densité du trafic et les événements spécifiques à l'intersection.
- **Agents de Véhicules :** Ces agents représentent les véhicules en circulation dans la zone de l'intersection. Ils transmettent des informations sur leur position aux agents de l'intersection. Cette sensibilité leur permet de réagir de manière adaptative aux changements de conditions, contribuant ainsi à une coordination fluide à l'intersection.
- **Agents de Piétons :** Les agents de piétons jouent un rôle essentiel dans la gestion de la sécurité à l'intersection. Ils détectent les signaux de traversée piétonne et collaborent avec les agents de l'intersection pour assurer un passage sûr, en tenant compte des variations du contexte, comme le trafic intense ou les conditions météorologiques défavorables. Ainsi la sensibilité au contexte à tous les niveaux, permet une prise de décision distribuée et adaptative. En ajustant les actions des agents en fonction du contexte spécifique, notre approche vise à améliorer non seulement la coordination et la sécurité à l'intersection, mais également à favoriser une fluidité accrue du trafic dans des conditions variées. Elle devient ainsi une pierre angulaire, assurant une gestion dynamique et efficace des intersections dans des environnements routiers en constante évolution.

4.2.1 Rôles d'agents

- **L'agent d'intersection (AI)** est chargé d'ajuster les horaires des feux de circulation et de collecter les données de base sur le trafic à partir du capteur de compteur de véhicules. Sur la base de ces données collectées, l'agent d'intersection décide s'il doit (1) continuer à utiliser les mêmes horaires d'éclairage, (2) les réduire pour une file d'attente au détriment de l'autre,

(3) le bloquer sur la couleur rouge pour permettre à un véhicule prioritaire pour traverser l'intersection, ou (4) laisser les piétons traverser la route chaque fois qu'ils y sont autorisés. L'agent d'intersection met à jour sa base de connaissances par des informations telles que la liste du trajet à vérifier, le nombre de véhicules dans les différentes directions, le mode (équitable, compteur, prioritaire).

- **Les agents de véhicules (AV)** sont des entités mobiles traversant l'intersection. Ces agents partagent la même structure mais ont des identifications et des données différentes. Les données encapsulées comprennent l'identifiant de l'agent, la destination, le type de véhicule (ordinaire, prioritaire) et pour les besoins de simulation l'heure d'arrivée et l'heure de sortie ainsi que le temps de séjour.
- **Les agents piétons (AP)** sont également des entités mobiles qui traversent la route. Comme les autres agents, ceux-ci ont la même structure. Leurs déplacements sont contrôlés par un feu piéton dont les couleurs sont toujours opposées à celles des feux des véhicules.

Il est à noter que ces trois types d'agents interagissent avec les feux tricolores (pour véhicules et piétons) et avec le compteur capteur du véhicule. Ceux-ci sont considérés ici comme de simples capteurs/actionneurs plutôt que comme des agents réactifs.

4.2.2 Interactions avec les agents

Le capteur de compteur de véhicules compte les véhicules des deux côtés de l'intersection et envoie le nombre de véhicules venant de l'est ou du nord à l'agent de l'intersection. En conséquence, l'agent de l'intersection peut sélectionner le mode approprié dans lequel les lumières s'allumeront, comme équitable, compteur ou priorité.

- **Équitable :** signifie adopter un certain intervalle de temps pour changer les feux de circulation, on y recourt dans le cas où le nombre de véhicules converge dans les deux sens.
- **Compteur :** c'est le cas dans lequel il y a une grande différence entre les véhicules venant des deux directions opposées. La priorité est ici donnée à la route encombrée et à chaque fois il y a un test jusqu'à ce qu'il ne reste plus aucun véhicule à l'intersection.
- **Priorité :** il est ici question de la priorité des véhicules tels que l'ambulance, les voitures de police et les pompiers. Ce type de véhicule a un facteur de temps très sensible, et tout retard entraîne des catastrophes. Ainsi, une fois détectés, ils ont eu la priorité lors du passage à l'intersection. Plus précisément, les agents des véhicules prioritaires informent l'agent de

l'intersection au moyen d'un message qui provoque le blocage des feux de couleur rouge dans l'autre sens permettant à la voiture prioritaire de traverser rapidement le carrefour. Les agents des véhicules peuvent communiquer entre eux afin d'entreprendre les actions appropriées en cas de congestion ou d'accident de la route et de cette manière, les véhicules changeront par exemple la destination initiale ou emprunteront un chemin alternatif (du nord à l'est ou vice versa).).

L'architecture proposée est donnée dans la figure 4.1 ci-dessous

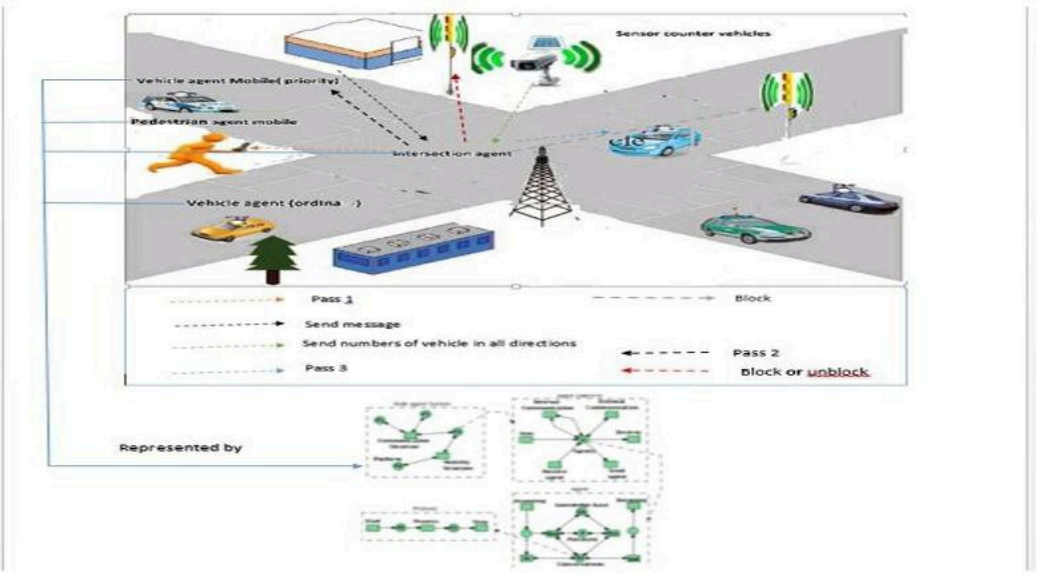


FIGURE 4.1 – L'architecture proposée

4.2.3 Modélisation du système multi-agent proposé

L'utilisation de réseaux dans les réseaux comme paradigme de modélisation permet l'utilisation directe de modèles de système au moment de l'exécution. Ceci peut être exploité comme suit : Le système global sera conçu comme un réseau de systèmes où les emplacements sont destinés à définir des lieux. Les transitions modélisent les déplacements possibles entre lieux ou la création d'agents (Figure 4.2). Chaque agent véhicule est créé à l'aide de l'inscription de transition vh :new Vh(Id, Sce, Dest, ta, ts, tr) et pour laisser ces agents passer par l'intersection, les transitions :mve(id), :mvee(id), : mvn(id) et :mvnn(id) sont utilisées pour déplacer respectivement un véhicule identifié par son identifiant sur la route Est ou sur la route nord. À noter que si l'on connaît l'origine des agents (la variable Sce), la destination (la variable Dest) est définie aléatoire-

ment. Autrement dit, un agent venant de la section Est peut se rendre soit vers l'ouest, soit vers le sud. Aussi, la création et le déplacement des agents piétons sont visibles sur le réseau principal. De tels agents sont créés au moyen de la transition inscrite par pg : new Pg(Pd) et leurs mouvements sont contrôlés par les transitions inscrites par : mv1(Pd) et :mv2(Pd). La figure 4.2 ci-dessous représente l'ensemble du système.

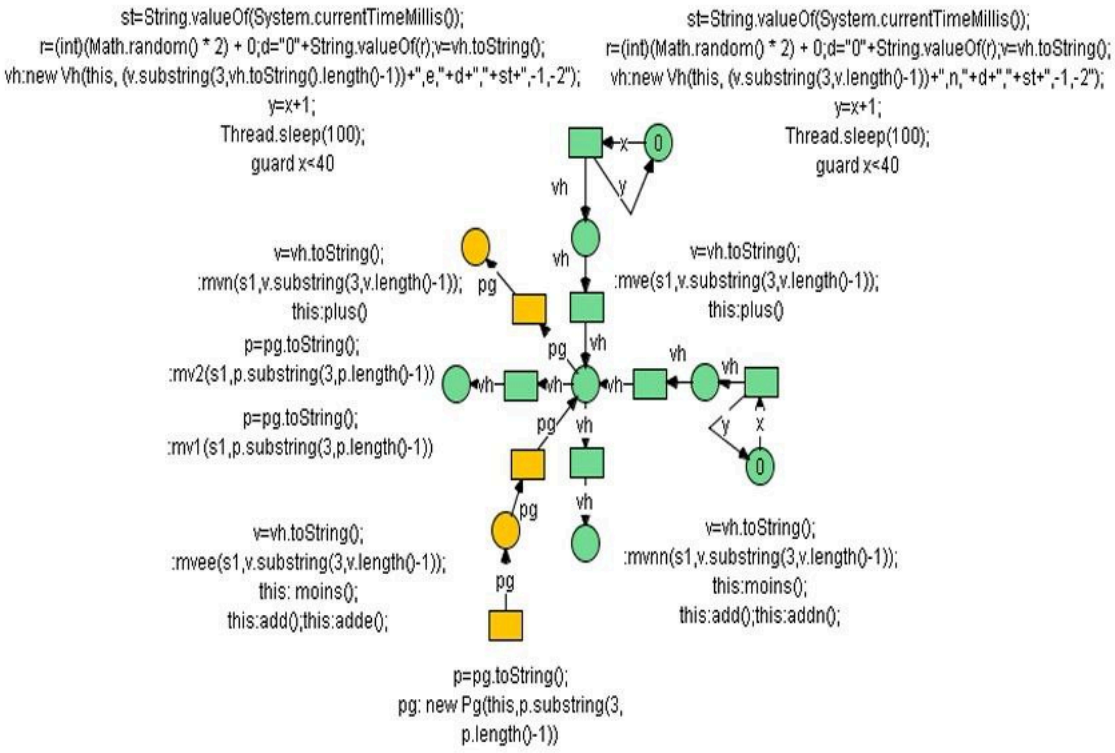


FIGURE 4.2 – Le réseau global du système.

FIGURE 4.3 – Les réseaux des capteurs du compteur du véhicule.

La figure 4.4 représente le réseau du système de feux de circulation. Les places centrales correspondent respectivement aux feux rouges et verts. Le symbole "1" indique un feu vert et le symbole "0" représente un feu rouge. À l'inverse, une place contenant le numéro opposé (couleur opposée)

est utilisée pour contrôler le feu tricolore pour les piétons. Les transitions annotées avec : north () et : east () permettent de changer l'état de la lumière en utilisant la formule y modulo 2. Autrement dit, elles déterminent la transition d'un état à un autre en fonction de la valeur actuelle de la lumière et en appliquant la formule y modulo 2. Les transitions : teste (y) et : testn(y) sont utilisées pour évaluer la valeur de la lumière sur la section est ou sur la section nord de la route. Ces transitions agissent comme des points de contrôle, permettant de tester l'état actuel et de prendre des décisions en fonction de la valeur de la lumière dans ces sections spécifiques de la route. De manière similaire, la transition : testep(y1) est utilisé pour contrôler le feu tricolore destiné aux piétons. La valeur renvoyée par cette évaluation est stockée dans la variable y, ce qui peut être utilisé ultérieurement dans le réseau pour prendre des décisions en fonction de l'état du feu tricolore pour les piétons. Enfin, les transitions : bloq1 () et : bloq2 () sont employées pour bloquer les feux de circulation, ce qui est représenté par le retrait de jetons des places centrales. Cette action de blocage est représentée graphiquement et peut être déclenchée par certaines conditions définies par le réseau.

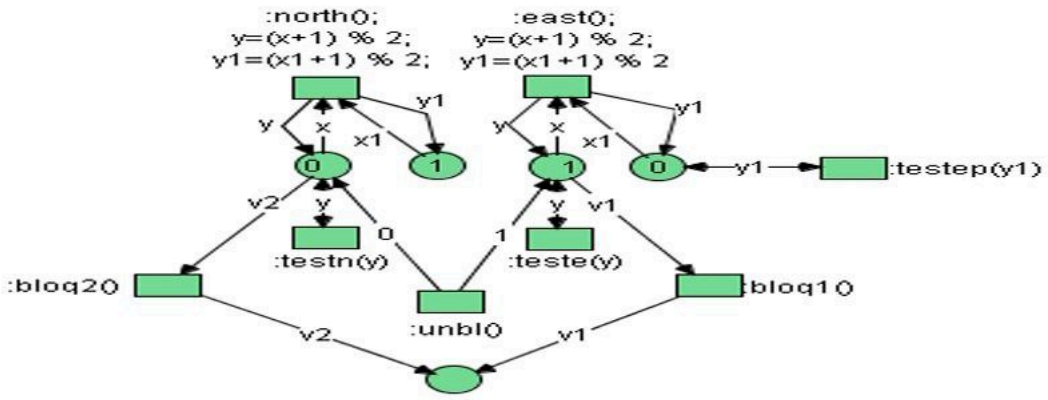


FIGURE 4.4 – Le sous-système de feux de circulation

Les agents mentionnés ci-dessus partagent la même structure représentée dans le côté inférieur droit de la figure 4.5. Les comportements de ces agents sont décrits en termes de protocoles. La sélection du protocole peut être effectuée de manière proactive ou réactive. Cette distinction se réfère à l'accès bilatéral au lieu qui contient les protocoles. La seule différence dans l'activation et l'apparition des transitions rea et pro est l'arc allant du lieu des messages entrants à la transition rea. Ainsi, il ne peut être activé que par les messages entrants. Chaque agent possède une base de connaissance qui contient des informations sur l'agent lui-même.

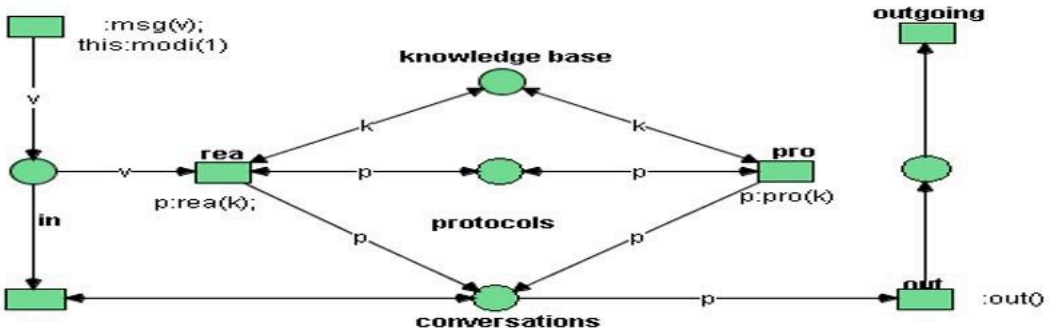


FIGURE 4.5 – structure de l'agent d'intersection

Dans les sous-sections suivantes, les protocoles et la base de connaissances des agents sont décrits. Commençons par l'agent d'intersection voir la figure 4.6. L'agent d'intersection commence par vérifier la présence d'un message provenant d'un véhicule prioritaire. Dès la détection d'un tel message, l'agent déclenche la transition : rea () sur le réseau principal, réagissant ainsi de manière spécifique. À l'inverse, en l'absence de tout message prioritaire, l'agent adopte une approche proactive en initiant la transition : pro () sur le réseau principal.

Dans le scénario de la seconde situation, la branche jaune, l'agent d'intersection joue un rôle crucial dans l'ajustement du signal lumineux en accord avec une période de temps spécifiée, fixée à 1000 ms selon notre modèle. Il est important de noter que cette adaptation se fait indépendamment de la priorité attribuée aux véhicules. Le processus s'initie par le vidage de la place centrale du réseau d'intersection, suivi par une modification simple des couleurs des feux de signalisation, le tout en conformité avec la durée prédéfinie. Cette démarche est caractérisée par une approche temporelle préétablie plutôt que par une considération de la priorité des véhicules dans la prise de décision.

Suite à cela, l'agent d'intersection procède à l'évaluation du nombre de véhicules dans chaque direction afin de déterminer s'il doit persévérer dans le mode actuel ou passer à un autre mode. En cas de disparité entre le nombre de véhicules sur différentes voies par exemple, si une voie compte davantage de véhicules que l'autre, l'agent d'intersection bloque le feu de couleur verte pour la voie qui présente une congestion plus importante (branche verte). Ce sous-protocole se répète de manière itérative jusqu'à ce que le nombre de véhicules atteigne zéro. Dans le scénario impliquant des véhicules prioritaires (branche rouge), l'agent d'intersection adopte une approche réactive face à l'arrivée d'un véhicule prioritaire, identifiable par un numéro d'identification d'agent multiple de

10. Dans cette situation, l'agent bloque les feux de circulation en activant les transitions m : bloq1() et m :bloq2(), permettant ainsi au véhicule prioritaire de traverser l'intersection de la manière la plus rapide possible. Une fois cette manœuvre accomplie, le mode normal est restauré grâce à la transition m : unbl ().

La structure du deuxième agent, l'agent véhicule, est présentée dans la figure 4.7, offrant une représentation visuelle de son fonctionnement.

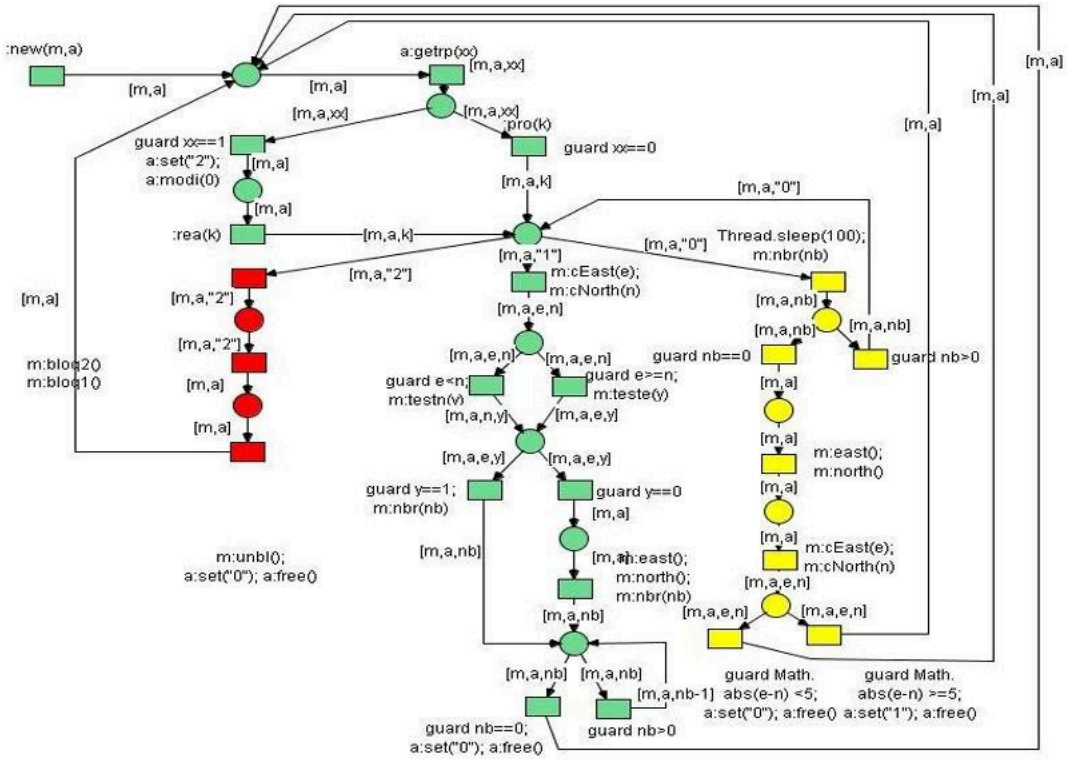


FIGURE 4.6 – Protocole de l'agent d'intersection

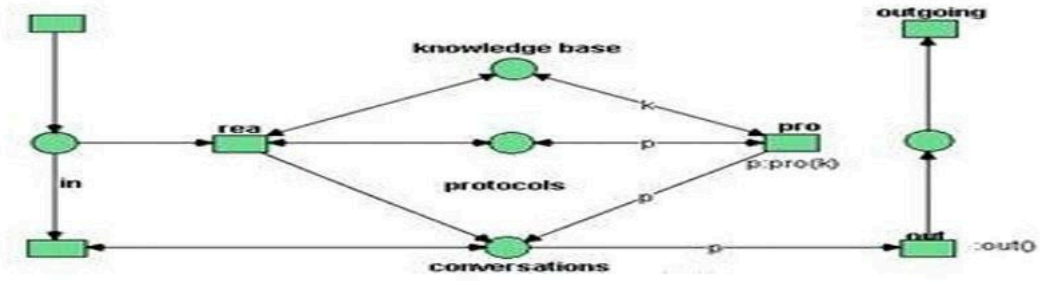


FIGURE 4.7 – La structure des agents véhicules et piétons.

L'agent de véhicule représente un ensemble prédéfini de véhicules circulant sur la route dans diverses directions. Une fois créés, ces agents adoptent une approche proactive en déclenchant la transition p : pro(k), qui instancie le protocole d'agent illustré sur la figure 4.7.

La partie droite du protocole (colorée en rouge) est spécifiquement dédiée au véhicule prioritaire, distinct des autres véhicules par l'identifiant multiple de 10. Lorsqu'un tel véhicule est présent, il envoie un message contenant un performatif (2) à l'agent d'intersection pour activer le mode prioritaire. Cela permet à ce véhicule de traverser l'intersection sans prendre en considération la couleur des feux de circulation.

En revanche, les véhicules normaux adoptent une approche différente. Ils doivent vérifier la signalisation lumineuse en invoquant respectivement des transitions notées sur le canal m : teste (1) ou m :testn (1) situé dans le protocole de l'agent véhicule. Cette méthode assure que le véhicule, en provenance de l'est ou du nord, peut traverser en prenant en compte la destination définie dans sa base de connaissances : sud (places/transitions roses) ou ouest (places/transitions jaunes). Ces vérifications permettent aux véhicules normaux de respecter les règles de circulation établies par le réseau de feux tricolores.

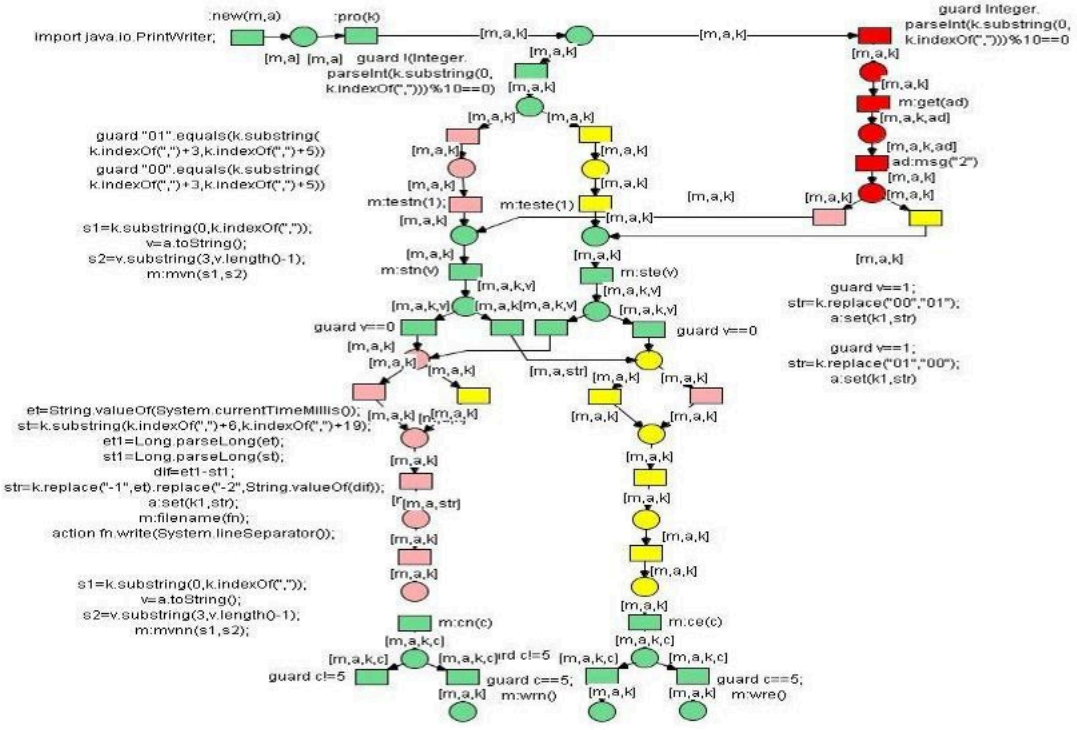


FIGURE 4.8 – Le protocole des véhicules.

Les véhicules qui pénètrent dans la place centrale, c'est-à-dire l'intersection, peuvent se heurter à des travaux sur la route, entravant ainsi leur progression. Dans de telles circonstances, l'agent en question informera immédiatement ses successeurs de cette perturbation potentielle. Ces agents ajusteront instantanément leur destination pour maintenir un flux de trafic fluide en tout temps. Les zones spécifiques du réseau qui gèrent de telles situations sont identifiables par la couleur verte, présente à la fois au centre et à l'extrémité de la figure 4.8.

Enfin, abordons la structure des agents piétons, telle qu'elle est représentée sur la figure 4.7. Ces agents partagent une structure similaire à celle des autres agents. Le protocole qui les autorise à traverser la route lorsque le feu est vert est détaillé sur la figure 4.9. Les agents piétons suivent un processus spécifique permettant de garantir leur sécurité tout en utilisant l'intersection. Ce protocole prend en compte l'état du feu de signalisation pour assurer un passage sécurisé des piétons lorsque le feu est favorable.

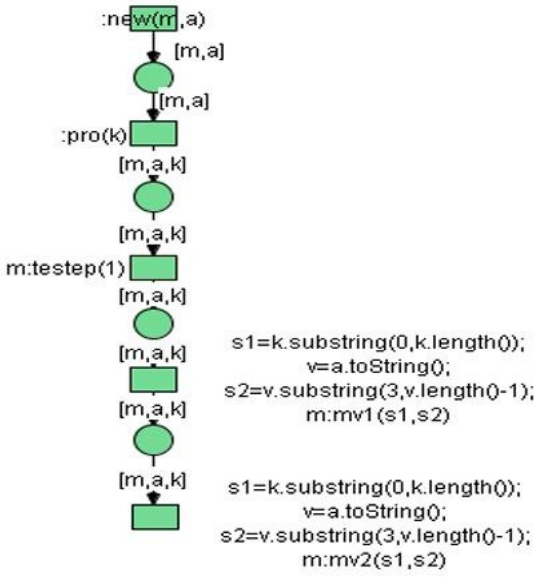


FIGURE 4.9 – Le protocole du piéton.

4.2.4 SIMULATION

La figure 4.10 offre un instantané d'un exemple de simulation du système proposé, fournissant ainsi une représentation visuelle de l'état du modèle à un moment donné. Le processus de simulation permet une analyse approfondie des performances du modèle dans divers scénarios. Les composants présentés dans cette illustration comprennent l'environnement, le feu tricolore, les véhicules, ainsi

que le protocole des agents d'intersection. Dans cet exemple, la création de véhicules est mise en évidence dans les deux directions, avec une hypothèse selon laquelle un véhicule apparaît dans la vue toutes les 2 unités de temps. De plus, la répartition des véhicules dans la vue est illustrée, avec 400 véhicules provenant du nord et 100 venant de l'est. Cette représentation offre une compréhension détaillée de la configuration spécifique de la simulation, mettant en lumière le flux de trafic et l'interaction entre les différents éléments du système.

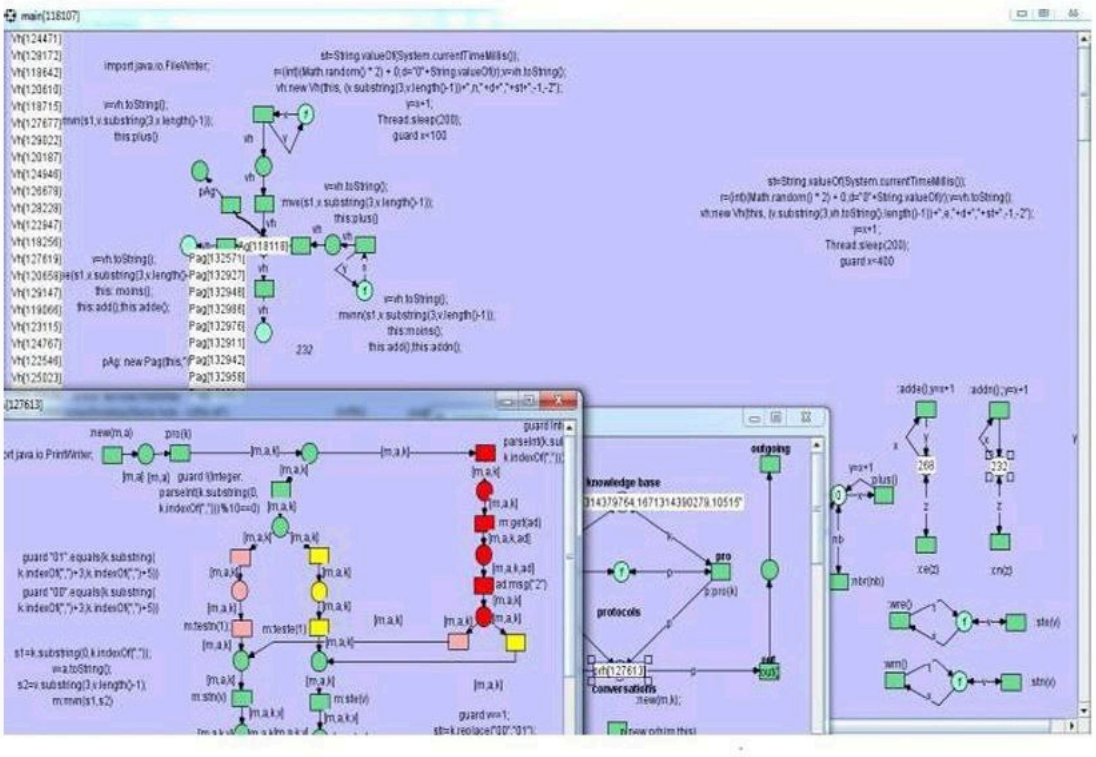


FIGURE 4.10 – Un exemple de simulation.

4.2.5 Résultats

La première colonne du tableau à quatre lignes ci-dessus est intitulée "Identifiant" et contient des nombres entiers aléatoires. Dans notre analyse, nous avons utilisé des symboles tels que N pour Nord et E pour Est pour indiquer l'emplacement des véhicules dans la deuxième colonne nommée "Source". La troisième colonne, "Destination", spécifie l'endroit vers lequel les véhicules se dirigent, permettant ainsi d'examiner deux scénarios : si le véhicule se déplace vers le sud (S) ou l'ouest (O). Le temps nécessaire aux véhicules pour atteindre leur destination depuis l'intersection est indiqué dans la dernière colonne, appelée "Temps de réponse".

En analysant les tableaux 4.1 et 4.2 susmentionnés, on peut noter que dans toutes les cases de couleur jaune, représentant les véhicules prioritaires, le temps d'attente à l'intersection est toujours significativement court par rapport au reste des véhicules qui les ont précédées, répondant ainsi aux exigences. En effet, pour ce type de véhicules, le temps est un facteur critique. Ces observations mettent en lumière l'efficacité du système pour gérer les véhicules prioritaires et minimiser leur temps d'attente à l'intersection, ce qui est essentiel compte tenu de leur nature prioritaire.

Identifiant	Source	Destination	Temp de réponse
26192	E	S	8244
26261	E	S	7025
26329	E	W	7353
26397	E	W	5777
26404	E	W	4155
26412	E	S	5367
26464	E	S	4961
26479	E	S	3589
26487	E	S	2700
26527	E	W	2793
26535	E	W	4619
26564	E	W	4557
26574	E	W	3292
26710	E	W	2981
27085	E	S	2485
27131	E	W	1077
27225	E	S	7396
26192	E	S	8244
26404	E	W	4155
26564	E	W	4557
26574	E	W	3292
26710	E	W	2981
27085	E	S	2485
27225	E	S	7396

TABLE 4.1 – Temps de réponse en intersection

Identifiant	Source	Destination	Temp de réponse
26201	N	S	7916
26215	N	S	8212
26233	N	W	6573
26241	N	W	5699
26272	N	W	6572
26285	N	W	7197
26309	N	S	6650
26590	N	S	3199
26664	N	S	2449
26777	N	S	3703
27061	N	S	3889
27138	N	W	5882
27200	N	W	1825
27310	N	W	6756
27342	N	W	4572
27369	N	S	3761
27391	N	S	4618
27431	N	W	6760
27536	N	W	5091
27559	N	W	3496
27566	N	W	3151

TABLE 4.2 – Temps de réponse en intersection (véhicules venant du nord)

4.2.6 Analyse

Pour garantir l'efficacité de notre approche, nous avons comparé nos résultats avec le Standard Traffic Light System (STLS) [70] et avec une autre approche qui utilise Adaptive Value Tracking (AVT) [12]. Les résultats sont résumés dans les figures 4.11 et 4.12.

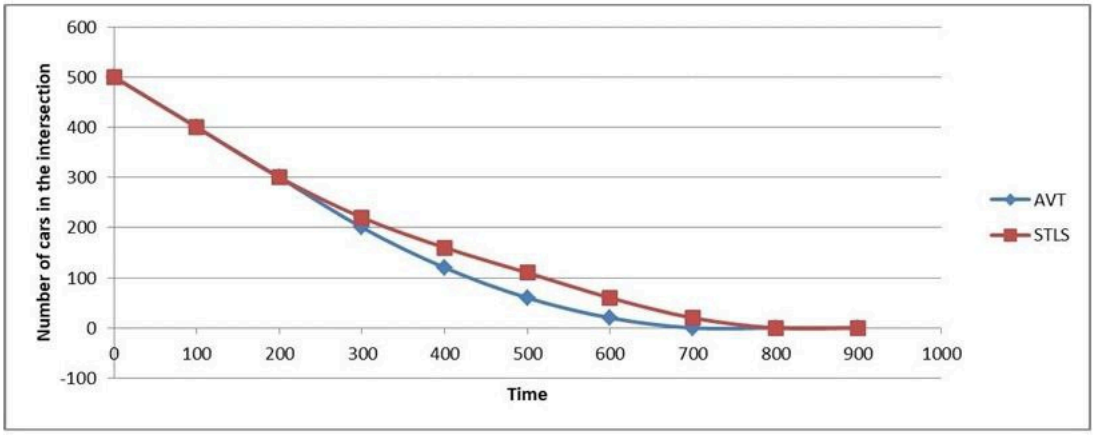


FIGURE 4.11 – réduction des embouteillages (méthode STLS, AVT).

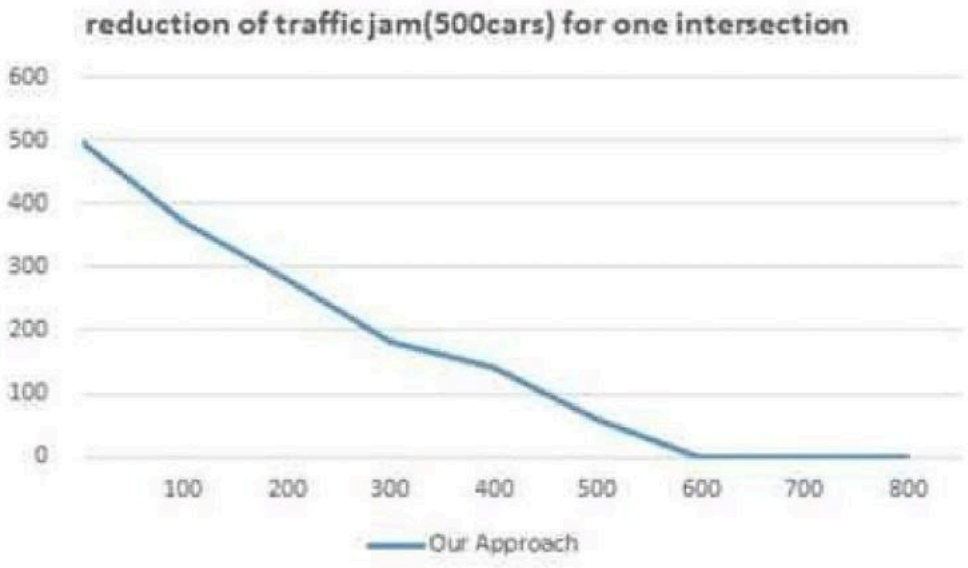


FIGURE 4.12 – réduction des embouteillages (notre approche).

STLS s'appuie sur des signaux lumineux qui fonctionnent à un temps fixe et prédéterminée qui ne change pas quelles que soient les circonstances. AVT a été conçu pour la communication I2I, où ils ont modélisé les intersections à l'aide de trois types d'agents : les agents de visualisation, les agents de feux de circulation et les agents d'intersection, un rôle spécifique est attribué à chacun d'eux.

Le scénario restreint mentionné ci-dessus est utilisé avec une seule intersection où l'on considère qu'il y a 500 véhicules dans l'intersection. Leur répartition est la suivante : 400 sur la place nord et

100 sur celle de l'est. On peut voir qu'en utilisant l'approche proposée, l'intersection devient vide après 600 unités de temps, mais en utilisant STLS, tous les véhicules passent dans une période estimée à 800 unités de temps. Avec l'approche utilisant l'AVT, il ne reste plus de véhicules dans l'intersection après 700 unités de temps.

Les deux approches évoquées ci-dessus n'ont pas pris en compte le cas des véhicules prioritaires, qui doivent être manipulés avec précaution par rapport aux autres véhicules car leur temps d'attente est sensible et il faut faire au mieux pour le rendre le plus réduit possible pour éviter de graves problèmes.

Dans notre approche, ce cas a été étudié et la solution proposée minimise le temps d'attente à l'intersection (le temps de réponse) par rapport aux autres véhicules tout comme surlignés en jaune dans le tableau 2, tableau 3.

Le système utilisé dans l'AVT ne traite pas le cas où le véhicule vient de l'est et se dirige vers le nord, ou vice versa. Il s'est également appuyé uniquement sur la communication (V2I) et n'a pas pris en compte la communication (V2V), dont l'avantage réside dans les situations où un grand nombre de véhicules affluent dans une direction particulière, rendant cette voie chargée. Pour résoudre ce problème, dans notre approche, nous avons utilisé une communication (V2V). Ainsi, le véhicule qui constate le problème de saturation communique avec les autres véhicules, qui prennent alors les mesures appropriées, qu'il s'agisse de rester dans la zone d'attente ou de changer de direction (ou de voie). De ces résultats, on peut conclure que le système réduira la taille des files d'attente et fournira des informations aux véhicules pour éviter les embouteillages et assurer une circulation fluide. Ainsi, notre système minimise également le retard des véhicules prioritaires.

4.3 CONCLUSION

Notre système de gestion du trafic se distingue par sa sensibilité au contexte, une caractéristique qui lui permet de s'adapter de manière dynamique aux conditions environnementales changeantes. Cette sensibilité au contexte est rendue possible grâce à l'utilisation de l'approche multi-agents et à la communication collaborative entre les agents intelligents du système. Voici les principaux éléments qui permettent à notre système de manifester cette sensibilité au contexte :

1. **Surveillance et perception du contexte :** Les agents du système sont équipés de capteurs et de mécanismes de perception qui leur permettent de surveiller en temps réel l'environnement routier. Cela inclut la détection en temps réel du nombre de véhicules présents sur la

route et la reconnaissance des véhicules prioritaires.

2. **Communication entre agents :** Les agents du système échangent activement des informations sur l'état du trafic et les conditions environnementales. Cette communication bidirectionnelle permet une collaboration en temps réel, où les agents partagent leurs observations et ajustent collectivement les décisions pour optimiser la gestion du trafic.

3. **Ajustement Dynamique des Feux de Circulation :** La sensibilité au contexte se manifeste dans l'adaptabilité du système à différentes situations. Par exemple, en présence de véhicules prioritaires signalés par un message spécifique, le système réagit en modifiant immédiatement les feux de circulation pour faciliter le passage rapide de ces véhicules, minimisant ainsi les temps d'attente.

Dans la continuité de notre recherche, nous reconnaissons la nécessité d'approfondir la représentation du contexte au sein des Systèmes Multi-Agents (SMA) appliqués aux véhicules intelligents. À cet égard, notre prochaine étape consistera à intégrer les règles JESS en complément de l'outil RENEW. Cette approche, en conjuguant la puissance des systèmes experts et la sophistication des SMA, permettra une représentation plus riche et nuancée du contexte urbain. Ainsi, notre transition vers cette méthodologie plus intégrée témoigne de notre engagement à améliorer la performance et l'adaptabilité des véhicules autonomes dans des environnements complexes.

Chapitre 5

Intégration des Règles JESS avec RENEW pour les Véhicules Intelligents

5.1 Introduction

Dans ce chapitre, nous progressons dans notre exploration des systèmes autonomes appliqués aux véhicules intelligents en adoptant une approche pour la gestion du trafic dans des environnements urbains dynamiques. Nous avons précédemment examiné la modélisation basée sur des agents mobiles pour résoudre les problèmes de congestion du réseau véhiculaire, utilisant principalement l'outil RENEW pour représenter la base de connaissances. Toutefois, afin de mieux répondre à la complexité croissante de ces environnements, nous avons élargi notre approche en intégrant les règles JESS avec RENEW.

Cette fusion d'outils nous permet de créer un système de gestion du trafic plus réactif et adaptatif, avec une sensibilité au contexte, permettant une meilleure gestion des variations en temps réel. En exploitant la communication collaborative entre les agents intelligents du système, nous sommes en mesure de surveiller en temps réel l'environnement routier, d'échanger des informations sur l'état du trafic et d'ajuster dynamiquement les décisions pour optimiser la gestion du trafic.

Nous débutons notre étude en expliquant les raisons qui ont motivé notre choix de cette approche. Ensuite, nous détaillons la représentation des divers agents, en explorant leurs rôles, avant de nous pencher sur la modélisation et la simulation. Enfin, la section conclusive présente la conclusion et la discussion.

5.1.1 Objectif et Motivation du travail

Notre approche de modélisation combine l'utilisation de l'outil RENEW [69] et les règles JESS (Java Expert System Shell). L'outil RENEW est utilisé pour décrire les hiérarchies naturelles au sein des systèmes d'agents et permettre la communication et la mobilité. Les règles JESS [38] revêtent une importance cruciale dans notre démarche. Ces règles expertes offrent une flexibilité exceptionnelle pour représenter et raisonner sur le contexte. En définissant des conditions et des actions associées, les règles JESS permettent aux agents mobiles de prendre des décisions informées en temps réel, en s'adaptant aux variations du contexte, ou d'autres facteurs environnementaux.

Les réseaux de référence [45] sont particulièrement appropriés pour la modélisation de systèmes sensibles au contexte dans les systèmes multi-agents, et ce pour plusieurs raisons. Tout d'abord, leur compatibilité avec l'architecture MULAN (Multi-Agent System Net within Net) [90] et l'outil RENEW les rend particulièrement efficaces. RENEW offre la possibilité d'appeler tout programme écrit dans le même langage, y compris les systèmes experts.

Un autre aspect à souligner est que les réseaux de référence illustrent clairement les principes fondamentaux des réseaux dans des réseaux.

5.2 Description de l'approche proposée

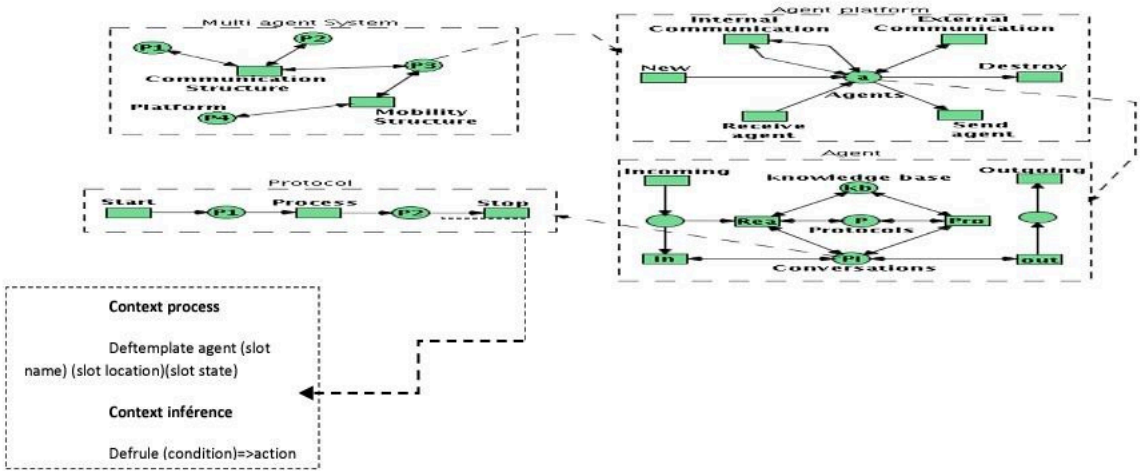


FIGURE 5.1 – Systèmes d'agents sous forme de réseaux dans des réseaux [44] modifiés

Dans le deuxième chapitre ainsi que dans le chapitre précédent, nous avons exploré la forme universelle de MULAN, comme illustré dans la figure 5.1. Cependant, notre approche dans cette étude diffère, car nous avons opté pour la représentation de la base de connaissances à l'aide de règles JESS plutôt que de sous-réseaux. Dans notre architecture, la construction de la base de connaissances de l'agent est réalisée grâce à JESS, un moteur d'inférence basé sur Java. JESS offre une flexibilité accrue dans la représentation des faits et des règles au sein de la base de connaissances. Cette alternative permet une modélisation plus précise et une gestion dynamique des connaissances, s'adaptant efficacement aux exigences spécifiques de notre système.

La base de connaissances elle-même est enrichie par l'inclusion de faits et de règles. Les faits constituent les données concrètes ou les observations pertinentes pour le système, tandis que les règles définissent la logique d'inférence permettant à l'agent de déduire des informations ou de prendre des décisions en fonction des circonstances.

L'utilisation de JESS comme base de connaissances offre des avantages significatifs, notamment une capacité d'inférence puissante, une gestion aisée des faits dynamiques, et une adaptabilité accrue aux évolutions du contexte. Cette transition vers JESS constitue ainsi une étape importante dans notre approche, renforçant la capacité de l'agent à raisonner, décider et agir de manière contextuelle au sein de l'environnement multi-agent.

Selon la structure architecturale évoquée précédemment, chaque étape joue un rôle crucial dans le processus de prise de décision au sein d'un système multi-agent. La transition entrante marque le point d'origine où le système commence à assimiler le contexte environnemental. Cette acquisition de contexte est ensuite transférée habilement à la transition **Rea**, qui agit comme un pivot central pour transmettre ces informations vers le protocole. Le protocole, en tant qu'entité de traitement, s'emploie à interpréter le contexte capturé et à déterminer l'action appropriée à générer en réponse. Un aspect remarquable de cette architecture est la manière dont les règles JESS interviennent lorsqu'un nouveau contexte émerge. Ces règles, en agissant comme des directives logiques, stockent ces nouveaux contextes dans la mémoire de travail, élargissant ainsi la base de connaissances du système. L'utilisation des constructions defrule dans le cadre de JESS constitue une étape essentielle. Ces constructions, s'appuyant sur les faits déjà affirmés dans un modèle def préalablement défini, permettent de déduire de manière intelligente l'action à entreprendre. Les résultats de ces déductions sont ensuite organisés et stockés dans un fichier texte selon les instructions spécifiées dans la figure 5.2. Cette gestion des résultats se réalise à travers la transition Pro, consolidant ainsi le lien entre la logique de déduction et l'exécution concrète d'actions. L'interaction finale

du système se matérialise par les transitions sortantes, qui gèrent la communication avec d'autres agents. Cette communication est essentielle pour coordonner les actions nécessaires à l'accomplissement des tâches spécifiques. En résumé, cette architecture propose une approche systématique et intelligente pour traiter les contextes changeants au sein d'un environnement multi-agent, en combinant l'acquisition de connaissances, la logique de déduction et la coordination des actions.

```
import java.io.FileReader;
import java.io.Filewriter;
import java.io.IOException;
import java.io.PrintStream;

import jess.*;

import java.util.Calendar;
import java.util.Date;
import java.util.Iterator;
import java.util.Vector;

public class car {

        public static void main(string[] args) throws JessException,
                        FileNotFoundException {

                try {

                PrintStream a = new PrintStream(new FileOutputStream("file path"));
                        System.setOut();
                        System.setErr(a);

                        Rete engine = new Rete();
                        engine.reset();

                        engine.batch("f");

                        Date d = new Date();
                        Calendar c = java.util.calendar.getInstance();
                        c.setTime(d);
                        int heure = c.get(java.util.calendar.DATE);
                        System.out.println(""+d);

                        ;


                        //ps.close();

                } catch (JessException e) {
                        e.printStackTrace();
                } /*catch (IOException ex) {
                        ex.printStackTrace();
                }*/

        }
}
```

FIGURE 5.2 – Code Java permettant le stockage des résultats

5.2.1 Un Aperçu Approfondi du Contexte Utilisant JESS dans les Systemes Multi-Agents

La base de connaissance de chaque agent, telle que représentée dans la Figure 5.1, connaît une amélioration substantielle grâce à deux processus distincts : le processus contextuel et l'inférence contextuelle.

• **Processus contextuel :** est en charge des informations contextuelles de l'agent. Elle est dé-

clenchée par de nouvelles perceptions de l'environnement mais aussi par des événements internes informant sur l'exécution des actions. Ici, nous nous concentrons sur le modèle basé sur des règles pour définir et représenter les informations contextuelles à un instant particulier t en fonction des données provenant des capteurs. Un système basé sur des règles est utilisé pour gérer les termes susmentionnés et permettre les fonctions d'ajout, de mise à jour ou de suppression de nouveaux faits. Ainsi, avec le modèle suggéré, nous pouvons décrire les informations contextuelles de l'utilisateur dans l'environnement informatique omniprésent. Ce modèle est efficace pour exprimer soit des informations contextuelles dynamiques, soit l'état environnemental. Nous pouvons également en déduire le statut de l'environnement contacté par l'utilisateur. Il est également prometteur de permettre à ce système de fonctionner automatiquement conformément à l'état déduit. Notez que chaque fait inclut un modèle. Le modèle a un nom et un ensemble d'emplacements, et chaque fait obtient ces éléments de son modèle.

- **Contexte d'inférence :** est responsable du processus d'inférence (également appelé raisonnement). Il est notifié par le processus contextuel afin qu'il puisse être sensible aux changements de contexte importants et déclencher un éventuel service ou action. Il est utilisé pour dériver de nouveaux faits basés sur les règles existantes dans les systèmes. Nous avons donc utilisé le même modèle mentionné précédemment pour réaliser cette étape. Pour déduire une situation basée sur JESS, nous classons chaque élément en fonction de ses caractéristiques et définissons une série de règles d'inférence. Si une situation se produit dans une ville par exemple, nous pouvons classer les contextes en plusieurs classes, comme l'appareil, les informations environnementales, le vehicule, le lieu, etc. Chaque classe possède des atomes. Par exemple, une personne de classe a des atomes tels qu'un conducteur. Ensuite, nous pouvons définir quelques règles JESS. Toutes les règles Jess sont définies à l'aide de la construction Defrule. Ils comportent donc deux parties, séparées par le symbole "=>". La première partie comprend le modèle LHS et la seconde comprend l'action RHS. Le LHS d'une règle est constitué de modèles utilisés pour faire correspondre les faits.

5.2.2 Modelisation de l'environnement

Un véhicule intelligent est une intégration complète de divers capteurs, modules de contrôle, actionneurs, etc. [93]. Un véhicule intelligent peut contrôler l'environnement de conduite, estimer les risques probables et prendre les mesures appropriées pour éviter ou diminuer le danger. Les

véhicules intelligents présentent un potentiel important pour aider les conducteurs à améliorer leur conscience de la situation et à réduire les erreurs. Ces véhicules sont équipés de systèmes automatiques avancés pour diverses fonctions et tâches préprogrammées. Ces systèmes, sont capables de collecter des données via des capteurs embarqués ou environnementaux, de convertir ces informations de détection en informations contextuelles de haut niveau et finalement d'entreprendre des actions basée sur ces informations contextuelles. La connaissance du contexte en voiture est extrêmement utile car elle peut aider les conducteurs à comprendre leur propre comportement de conduite, améliore l'efficacité énergétique en fournissant des itinéraires moins fréquentés, permet aux autorités de gérer les places de stationnement et régit les règles de circulation dans les zones urbaines à forte densité de population.

Pour assister et offrir le confort au conducteur avec une conduite plus facile, moins de charge de travail et moins de risques de blessure, un vehicule intelligent doit être capable de détecter, d'analyser, de prédire et de réagir à l'environnement routier, ce qui est la caractéristique de base de véhicule intelligent : sensible contexte. En ce sens et dans le cas de l'approche proposée, nous avons utilisé deux types d'éléments de contexte ; un nom descriptif doit être attribué aux applications à utiliser et pour fournir une compréhension convenue des contextes. Comme indiqué ci-dessous :

- **Contextes environnementaux :** Les contextes environnementaux sont alliés à l'environnement physique. Dans le véhicule intelligent, il comprend la description de l'état météorologique, l'état de la surface de la route, les informations routières, les panneaux de signalisation, le feu de signalisation et l'état du réseau.
- **Contextes vehicule :** le contexte vehicule comprend soit le système électrique, soit le système de sécurité, soit le système de confort. Le système d'alimentation concerne l'état du moteur (On ; Off), l'accélérographe, l'identification et la destination. Le système de sécurité comprend des fonctionnalités affectant la sécurité de la voiture et du conducteur, telles que l'état de l'airbag, la ceinture de sécurité, etc. Le système de confort concerne le détournement, la climatisation et les fenêtres.

5.2.2.1 "Scénario : Une Conduite Intelligente et Sensible au Contexte avec un Véhicule Connecté"

Après une journée épuisante au travail, un professionnel se dirige vers sa maison depuis le lieu A à bord de son véhicule intelligent, programmé pour réagir de manière contextuelle aux conditions routière et météorologique. Pendant son trajet vers le lieu Q, il traverse plusieurs segments routiers,

chacun présentant des caractéristiques uniques.

Le premier tronçon est une autoroute où il reçoit une notification de la station-service locale à la Place F, proposant un nettoyage gratuit de son véhicule à chaque ravitaillement complet en carburant. Son véhicule, conscient de son niveau de carburant, prend en considération cette offre.

Ensuite, il entre dans un village où la vitesse est limitée à 40 km/h. Après quelques minutes de conduite, il détecte des travaux sur la route suivante et une déformation de la chaussée à la Place K. Le véhicule ajuste automatiquement sa vitesse et son itinéraire en réaction à ces conditions, assurant la sécurité du conducteur et des passagers. Poursuivant son chemin dans une rue urbaine, à quelques kilomètres de sa destination finale, le véhicule détecte la présence d'un enfant sur la route à la Place P. Il réduit sa vitesse, prenant en compte la situation particulière où l'enfant court après un ballon.

Dans ce scénario, le véhicule démontre une compréhension avancée du contexte et de la sensibilité au contexte. Elle adapte sa vitesse en fonction des conditions routières, s'arrêtant ou changeant de direction en présence d'obstacles. De plus, elle prend des décisions basées sur le niveau de carburant pour bénéficier des offres pertinentes, tout en maintenant un équilibre optimal entre efficacité énergétique et fonctionnalités de confort. Cette approche informatique intelligente garantit une expérience de conduite sécurisée et fluide, tout en tenant compte des offres et des situations spécifiques rencontrées sur la route.

5.2.2.2 Représentation Dynamique des Véhicules et des Positions sur la Route"

Le véhicule et les différentes positions sur la route seront représentés à l'aide d'agents. L'utilisation de réseaux de systèmes comme paradigme de modélisation permet une utilisation directe des modèles de système pendant l'exécution. Le système global sera conçu comme un réseau de systèmes, où les lieux définissent des emplacements, et les transitions représentent les déplacements possibles entre les lieux ou la création d'agents (voir Figure 5.3). Pour une meilleure clarté, seulement un sous-ensemble de services sera présenté ici pour mettre en évidence le concept de modélisation. De plus, les détails concernant les inscriptions d'arc, qui permettent aux agents mobiles de se déplacer d'un endroit à un autre sont omis.

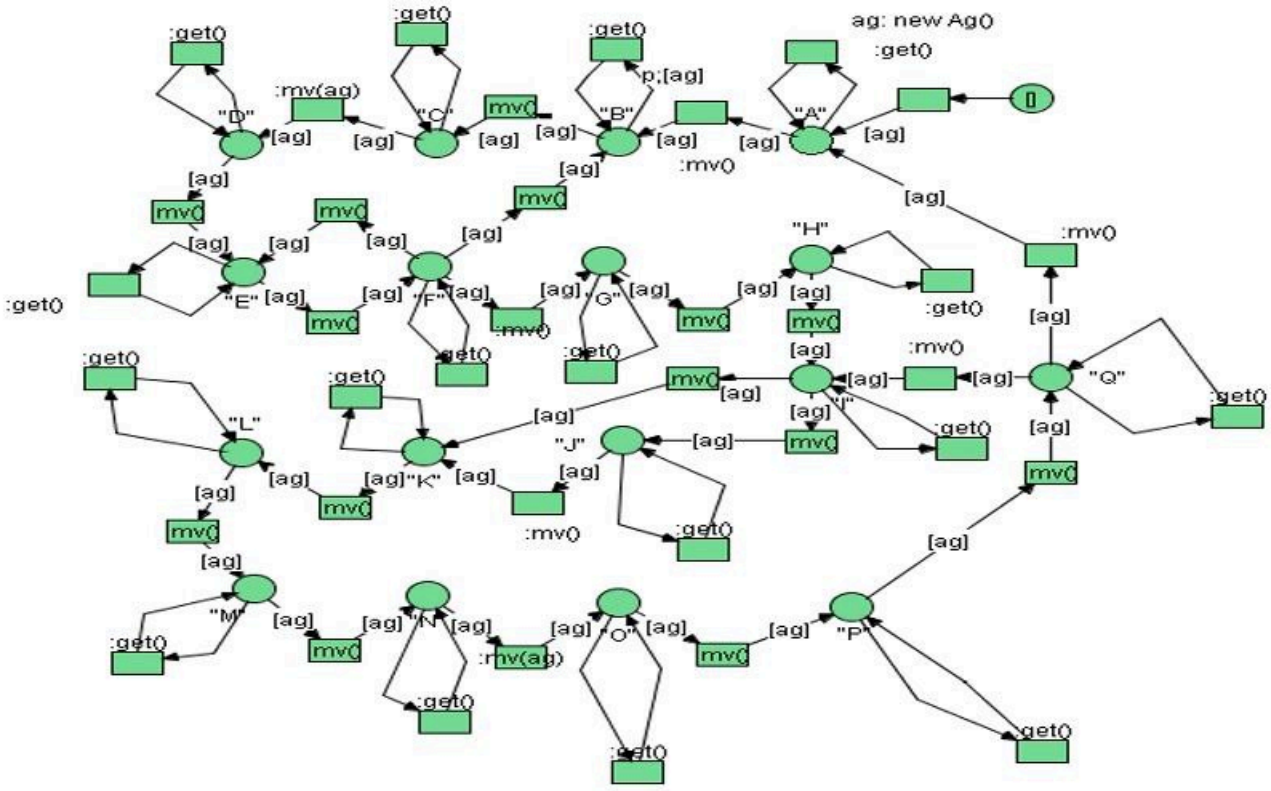


FIGURE 5.3 – Le système global

Dans cette approche, nous avons mis en place un système où des agents stationnaires cognitifs ont été attribués à des groupes de lieux, tels que des routes ou des plateformes. Ces agents, nommés plat1, plat2 et plat3, sont des entités responsables de lieux spécifiques, chacun étant doté d'une liste d'informations contextuelles relatives à ses services respectifs. L'agent plat1 est en charge du lieu situé entre les points A et D, qui correspond à une autoroute. Il est spécifiquement dédié à la gestion des services associés à cet itinéraire. De manière similaire, l'agent plat2 supervise le lieu entre les points E et G, qui représente une zone villageoise. Ses responsabilités incluent la coordination et la gestion des services spécifiques à cette localité. Enfin, l'agent plat3 prend en charge le lieu entre les points H et Q, correspondant à une zone urbaine. Sa mission consiste à gérer les divers services et informations contextuelles propres à cet environnement. Chaque agent stationnaire cognitif détient une liste exhaustive d'informations contextuelles relatives à son secteur d'affectation. Ces informations englobent divers aspects liés aux services, aux besoins locaux et aux particularités propres à chaque lieu. À côté de ces agents stationnaires, nous modélisons le véhicule

comme un agent mobile. Ce dernier interagit de manière dynamique avec les agents stationnaires cognitifs, permettant ainsi une communication et une coordination fluides entre les lieux assignés et la voiture en déplacement. Cette approche vise à optimiser la gestion des services en fonction des caractéristiques spécifiques de chaque région, assurant ainsi une expérience globale efficace et personnalisée.

Les agents mentionnés partagent une structure commune, représentée dans le coin inférieur droit de la figure 5.1. Dans ce scénario, le comportement de ces agents est défini en termes de protocoles, lesquels régissent leurs interactions. La sélection du protocole peut fondamentalement être effectuée de manière proactive ou réactive, en fonction d'un accès bilatéral au lieu qui détient ces protocoles. La particularité de cette approche réside dans la manière dont les protocoles sont déclenchés. La distinction entre la sélection proactive et réactive se manifeste par l'accès bilatéral au lieu où sont définis les protocoles. La seule différence notable réside dans l'activation et l'apparition des transitions "rea" et "pro", matérialisées par un arc reliant le lieu des messages entrants à la transition "rea". En d'autres termes, le choix entre un comportement proactif et réactif dépend de la nature des interactions avec le lieu en question. Lorsqu'un agent reçoit des messages entrants, l'activation de la transition "rea" est conditionnée par la présence d'un arc spécifique reliant le lieu des messages entrants à cette transition. Ce mécanisme offre une flexibilité dans la manière dont les agents réagissent aux stimuli extérieurs et initient des actions en conséquence. Ainsi, cette architecture permet une gestion adaptative des protocoles en fonction du contexte, favorisant une communication efficace entre les agents et les lieux, tout en offrant la possibilité de prendre des initiatives proactives lorsque nécessaire. Cette approche stratégique vise à optimiser la réactivité et l'efficacité globale du système dans des environnements dynamiques. Ainsi, il ne peut être activé que par les messages entrants. Les informations contextuelles font partie de cette base de connaissances d'agent qui peut être implémentée dans un simple de cas, par exemple sous forme de sous-réseaux. Ceci est vrai dans le cas des capteurs (modélisés comme des agents réactifs). Malheureusement, pour les agents cognitifs, cela conduit à un modèle fermé, qui doit souvent être entièrement repensé avant d'introduire quelques changements.

5.2.2.3 Caractéristiques des Agents et Contextes

Introduisons le "Tableau des Caractéristiques et Contextes des Agents" pour examiner plus en détail les détails spécifiques relatifs à chaque agent :

Agents	Type	Contexte
Vehicule	Mobile Intelligente	Identification=a Destination=Q Statut du moteur = Activé
Plat1(Autoroute)	Stationnaire Cognitive	bonnes routes Aucun obstacle nettoyage gratuit
Plat2 (village)	Stationnaire Cognitive	malformation dans la rue
Plat3 (Zone urbaine)	Stationnaire Cognitive	Enfant dans la rue

TABLE 5.1 – Caractéristiques et Contextes des Agents

En analysant ce tableau, on observe que chaque agent possède des caractéristiques distinctes, reflétant son rôle spécifique dans le système. Par exemple, la voiture mobile est intelligente, avec une destination définie (Q), tandis que les agents stationnaires cognitifs ont des contextes variés liés à leurs lieux respectifs.

Explorons maintenant l'utilisation de JESS en tant que modèle de base de connaissances pour les agents véhicules. En ce qui concerne l'établissement de la communication entre les agents, nous définissons le modèle de message ACL de la manière suivante :

(deftemplate ACL- Message (slot communicative-act) (slot sender) (slot receiver) (slot conversation-id) (slot protocol) (slot language) (slot content)).

Il est essentiel de noter que l'environnement dans lequel le véhicule évolue est étroitement lié à la route qu'il parcourt. La mémoire de travail de l'agent, agissant comme une sorte de référentiel dynamique, enregistre les informations contextuelles spécifiques à chaque trajet. Les modèles détaillés ci-dessous décrivent de manière approfondie les caractéristiques distinctes du contexte du véhicule en fonction de la route empruntée :

(deftemplate car (slot (identification) (slot luminosity) (slot music) (slot position) (slot destination) (slot action) (slot speed)(slot weather)(slot fuel)(slot engine-statue)(slot state-road)(slot content-road)(slot offer -platform))

L'assignation de valeurs spécifiques aux emplacements aboutit à la formation d'une entité plus

exhaustive et détaillée. Cette démarche permet d'enrichir les caractéristiques de chaque emplacement en attribuant des paramètres spécifiques, contribuant ainsi à une représentation plus complète et nuancée de ces lieux dans le cadre du système.

```
(assert (car (luminosity "low") (position village) (identification a)(destination Q)
          ( engine- statue  on) ( speed raised )(content - road  kid)).
```

Nous présentons à présent plusieurs règles JESS qui traduisent des changements de contexte. Dans la première règle, dès que la voiture manifeste l'intention de se diriger vers un village, une zone urbaine ou une autoroute, elle enregistre le contexte associé. En cas de détection d'un enfant ou d'un objet sur ces itinéraires, la règle active le frein de la voiture. En ce qui concerne la deuxième règle, lorsque le réservoir de carburant atteint un niveau critique, la voiture réagit en acceptant une offre proposée par la plateforme.

```
1: (defrule A
(car{(position == village)// (position == highway)//( position == urban area)} &&
 (content_ road == kid))
=>
(assert (ACLMessage (communicative-act INFORM) (sender platform) (receiver car)
 (content "kid") (conversation-id "information") ))(assert (car(braking )(engine
  statue off))
2: (defrule B
(car {(offer_ platform == free cleaning)&&( levelfuel==full)})
=>
(assert (ACLMessage (communicative-act INFORM) (sender platform) (receiver car)
 (content "free cleaning") (conversation-id "information") ))
(assert (car (action don't pass)))
```

De plus, la règle suivante prévoit une diminution de la vitesse lorsqu'une malformation de la route est détectée (pluie ou travaux de construction). defrule C

```
(car (state_ road == malformation) && {(position == village)// (position == highway)
//( position == urban area)}     (assert (ACLMessage (communicative-act INFORM)
(sender platform) (receiver car) (content "malformation in the road") (conversation-id
```

```
  "information") ))
          (assert (vehicule(speed decrease)))
```

En outre, la vitesse du véhicule varie en fonction de sa position géographique, atteignant une vitesse de 120 km/h sur l'autoroute, puis diminuant à 40 km/h une fois qu'il pénètre dans une zone urbaine.

```
 (defrule D
(car {(state_ road == good) && (position == highway) )
=>
(assert (ACLMessage (communicative-act INFORM) (sender platform) (receiver car)
 (content "speed is 120km/h") ))
(assert (car (speed 120km/h)))
(defrule E
(car {(state_ road == good) && (position == urban area) )
=>
(assert (ACLMessage (communicative-act INFORM) (sender platform) (receiver car)
 (content "speed is 40km/h") ))
(assert (car (speed 40km/h)))
```

La structure de l'agent véhicule est illustrée à la Figure 5.4.

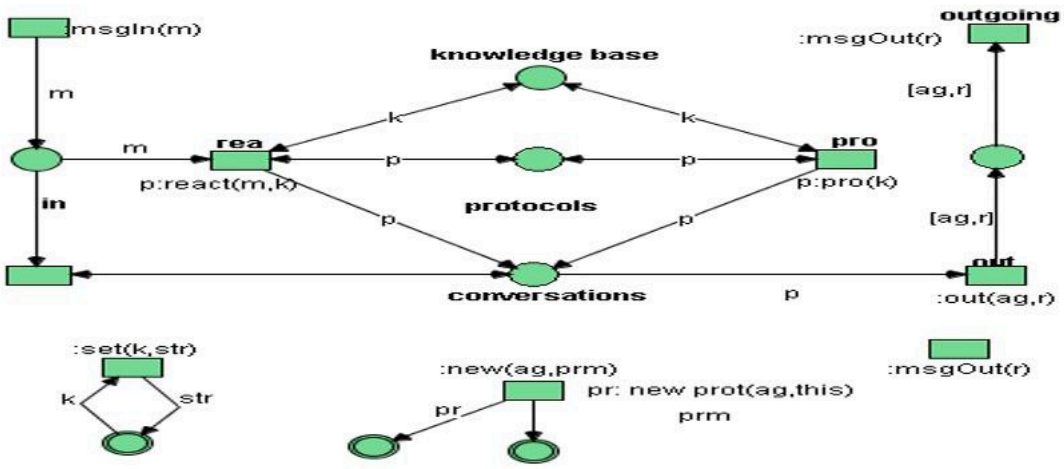


FIGURE 5.4 – L'agent véhicule

L'information est systématiquement transmise à l'agent véhicule pour qu'il puisse réagir de manière adaptative aux variations des données. En fonction de ces données, un protocole réseau évaluera de manière délibérée les actions les plus appropriées à prendre.

Le protocole encadrant le déplacement d'un véhicule d'une position à une autre tout en permettant la modification de son état de "Activé" à "Désactivé", représenté par les valeurs 0 ou 1, repose sur des transitions assorties d'expressions de garde. Une transition particulière ne peut être déclenchée que si toutes ses expressions de guard sont évaluées comme étant vraies. Ceci est clairement illustré dans la figure 5.5.

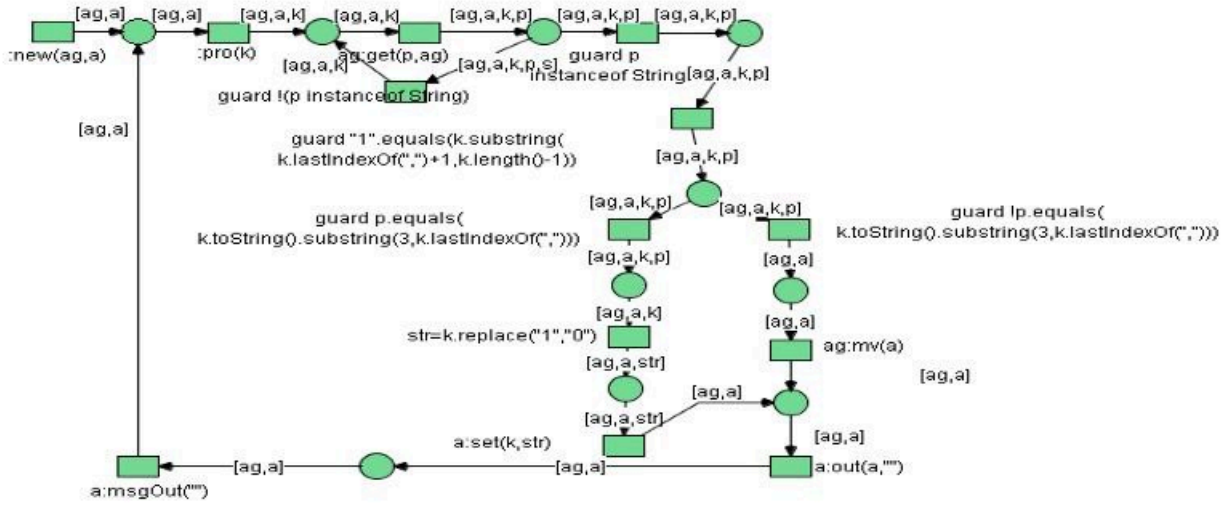


FIGURE 5.5 – Les protocoles de véhicule

5.2.3 Simulation

La figure 5.6 dépeint l'état d'un exemple de simulation dans notre système d'agent. Ce processus de simulation a pour objectif d'analyser les performances du modèle. Il introduit les caractéristiques générales du cadre de simulation, mettant en avant l'environnement, le véhicule en tant qu'agent, les faits JESS et le protocole spécifique au véhicule.

L'environnement de simulation englobe quatre agents, comprenant l'agent de véhicule et trois plateformes identifiées comme plat1, plat2 et plat3. Notre système de simulation propose plusieurs scénarios, et la figure suivante illustre les résultats de la simulation liés au scénario actuellement discuté. La fenêtre en arrière-plan offre une vue détaillée des faits et des règles JESS qui constituent la base de connaissances, c'est-à-dire le contexte, de l'agent.

En termes plus détaillés, l'environnement se compose de divers acteurs, tels que le véhicule et les plates-formes, qui interagissent selon des règles spécifiées par le cadre de simulation. Les faits et les règles JESS, visibles dans la fenêtre en arrière-plan, jouent un rôle crucial en définissant les éléments du contexte dans lequel l'agent opère. Le protocole de véhicule, quant à lui, guide les actions spécifiques de l'agent en réponse aux conditions définies par les règles de simulation.

Cette approche de simulation permet une évaluation approfondie des performances du modèle dans des scénarios variés, offrant ainsi des perspectives utiles pour l'optimisation et l'amélioration continue du système d'agents.

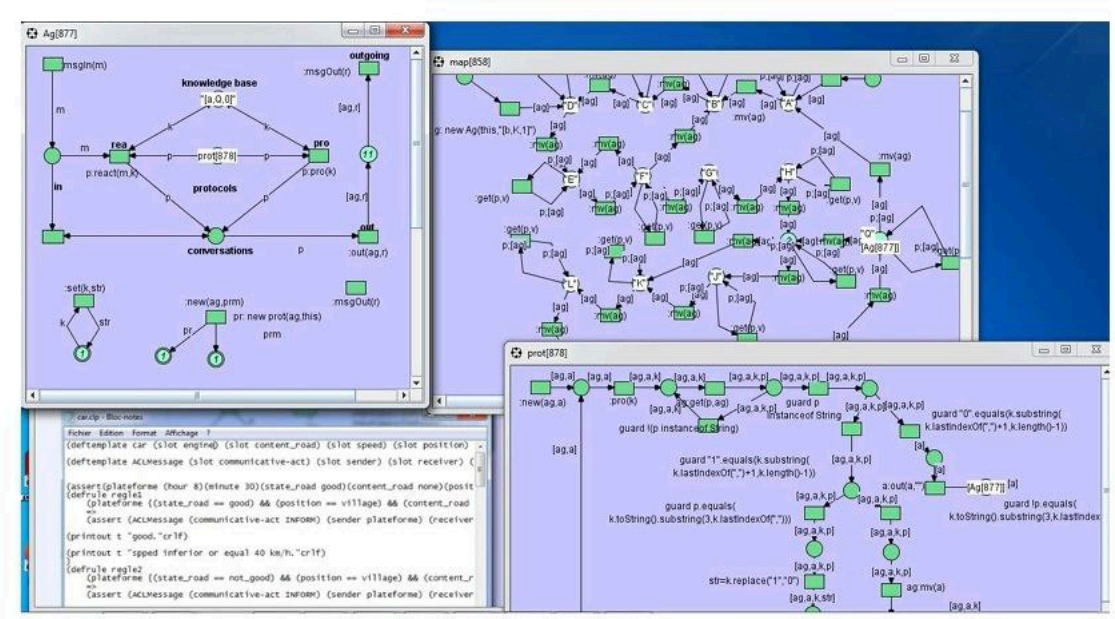


FIGURE 5.6 – Exemple de simulation

5.2.4 Exécution et stockage

Eclipse [13, 25] est un projet développé par la fondation Eclipse, organisé en un ensemble de sous-projets de développement logiciel. Il vise à créer un environnement de développement de logiciels libres extensible, universel et polyvalent, s'appuyant principalement sur Java. Nous avons utilisé le plugin Eclipse car cette plateforme est conçue et construite pour répondre aux besoins suivants : ne pas être limitée à un domaine particulier, faciliter l'intégration d'outils à travers divers types de contenu et fournisseurs d'outils, et permettre la création d'une variété d'outils pour le développement d'applications.

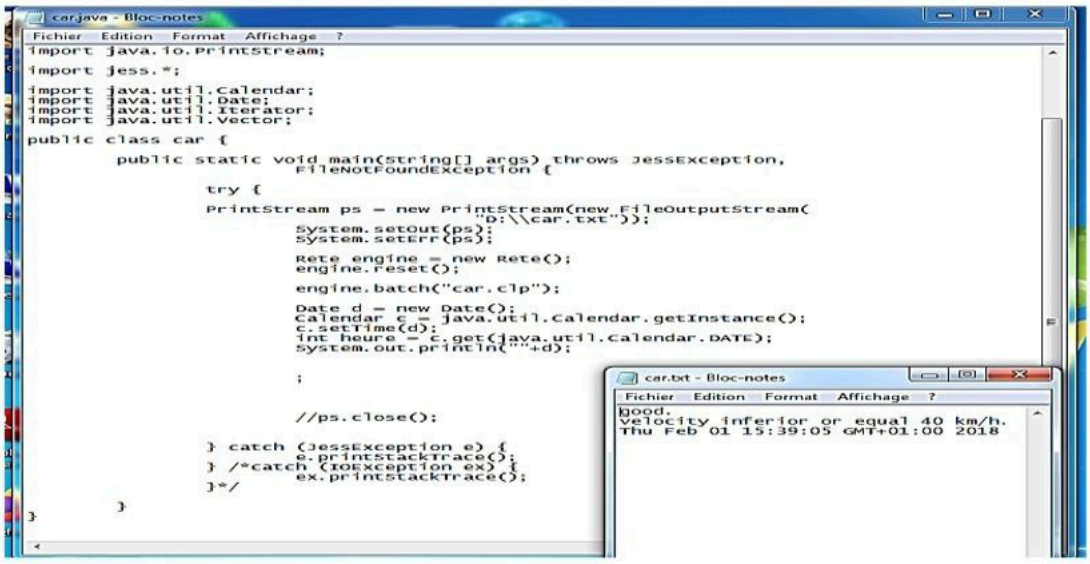


FIGURE 5.7 – Exécution et stockage

5.2.5 Discussion

L'objectif fondamental de cette recherche était d'établir un système hautement réactif aux contextes changeants. Plusieurs éléments suggèrent que le choix d'un système expert était judicieux pour représenter de manière concise ces contextes dynamiques. En outre, il est manifeste que JESS présente des avantages notables par rapport à Java en ce qui concerne la représentation des connaissances, avec une intégration aisée dans d'autres applications Java Web. Un autre avantage significatif réside dans les performances optimisées du moteur central de JESS, assurant une exécution rapide.

Cette recherche a mis en évidence la nécessité d'une synergie entre les outils JESS et RENEW pour concrétiser efficacement la mobilité et la communication au sein du système. La combinaison de ces deux outils offre une approche complète pour créer un environnement intelligent capable de s'adapter de manière dynamique aux situations changeantes, tout en facilitant une communication fluide entre les agents du système. Cette approche combinée tire parti des forces respectives de JESS et RENEW pour répondre aux défis de la mobilité et de la sensibilité au contexte de manière intégrée et performante.

En comparaison avec les approches discutées dans les précédents chapitres consacrés au travail connexe, notre approche se distingue par son fondement sur un modèle formel, à l'opposé des approches semi-formelles qui reposent sur UML. Cette proposition repose sur une extension des

réseaux de Pétri conventionnels, y compris ceux ordinaires, colorés et objets, vers un paradigme plus robuste connu sous le nom de ”réseaux dans les réseaux”, offrant ainsi une alternative aux approches centrées sur les réseaux de Pétri traditionnels.

L'architecture MULAN, basée sur ce paradigme, s'intègre élégamment au principe de composition des systèmes multi-agents. Dans le cadre de notre approche, cette architecture est renforcée par l'intégration du moteur de raisonnement JESS, inspiré des concepts des approches orientées aspect. Cette combinaison confère une adaptabilité fluide des agents au sein du système, accentuant ainsi la flexibilité et l'efficacité de la modélisation.

Notre choix d'un modèle formel, enrichi par les réseaux dans les réseaux, offre une approche expressive et puissante pour représenter et composer des systèmes multi-agents. L'incorporation de JESS injecte une logique de raisonnement avancée dans notre architecture, améliorant ainsi la qualité de la modélisation et la capacité à réagir aux contextes changeants de manière agile et précise.

5.3 Conclusion

En conclusion de ce chapitre, l'intégration réussie des règles JESS avec l'outil RENEW a ouvert des perspectives nouvelles et stimulantes pour les systèmes multi-agents mobiles, notamment dans le domaine des véhicules intelligents. La combinaison de la puissance expressive de JESS et de la flexibilité de RENEW a permis de concevoir des modèles plus complexes et adaptatifs.

L'étude de cas sur la sensibilité au contexte dans les systèmes multi-agents mobiles, en particulier avec les véhicules intelligents, a démontré comment ces technologies peuvent travailler en tandem pour améliorer l'efficacité et la réactivité des systèmes. L'approche de Renew, centrée sur les réseaux, s'est harmonieusement intégrée avec les règles JESS, offrant ainsi une solution complète pour la modélisation et la gestion des systèmes multi-agents mobiles.

Conclusion générale et travaux futurs

Au cours de cette thèse, nous avons proposé des systèmes multi-agents sensibles au contexte, visant à répondre aux défis complexes liés à la gestion du trafic urbain. Nous avons développé des modèles qui permettent aux systèmes de s'adapter aux variations du contexte opérationnel, particulièrement dans le cadre de la gestion du trafic aux intersections. Cette approche a permis de proposer une gestion dynamique et réactive des flux de véhicules, en optimisant l'efficacité du système dans des environnements urbains complexes.

Dans une deuxième approche, l'intégration des règles JESS avec l'outil RENEW a amélioré la réactivité et la gestion des systèmes multi-agents dans les véhicules intelligents. Cette intégration a permis une vérification formelle des interactions entre les agents, assurant des prises de décisions plus robustes et adaptées aux conditions spécifiques des intersections. Elle a renforcé la capacité du système à réagir de manière plus fluide et plus rapide aux variations du trafic et aux priorités des véhicules.

Pour garantir l'efficacité de notre approche, nous avons comparé nos résultats avec ceux obtenus par des systèmes existants, comme les systèmes de gestion de la circulation basés sur des feux de signalisation à temps fixe et les systèmes utilisant des méthodes d'adaptation. Nos résultats montrent que, dans un scénario simulé avec 500 véhicules à une intersection, notre approche permet de vider l'intersection après seulement 600 unités de temps, tandis que les autres approches nécessitent jusqu'à 800 unités de temps. De plus, notre système prend en compte les véhicules prioritaires, réduisant leur temps d'attente à l'intersection, ce qui est crucial pour assurer une circulation fluide et éviter des problèmes graves.

Notre approche inclut également l'utilisation de la communication entre véhicules (V2V), ce qui permet aux véhicules de s'ajuster en temps réel aux conditions de circulation, comme la saturation

d'une voie, et de prendre les mesures appropriées pour éviter les embouteillages. Ces résultats démontrent que notre système est non seulement plus rapide et réactif que les autres approches, mais qu'il minimise également le retard des véhicules prioritaires et optimise la gestion du trafic aux intersections.

Les perspectives et les travaux futurs dans le domaine des systèmes autonomes sensibles au contexte sont vastes et prometteurs. Voici quelques axes de développement qui pourraient être explorés :

1. Optimisation des algorithmes de traitement du contexte : Les futurs travaux pourraient se concentrer sur le développement d'algorithmes plus efficaces pour traiter et interpréter les informations contextuelles. Cela pourrait inclure des méthodes d'apprentissage automatique avancées, telles que le deep Learning, pour permettre aux systèmes de s'adapter de manière plus précise et dynamique à différents contextes.
2. Exploration de nouveaux domaines d'application : Il serait intéressant d'explorer comment les systèmes autonomes sensibles au contexte pourraient être appliqués à de nouveaux domaines, tels que la médecine, l'agriculture ou la gestion des ressources naturelles. Cela ouvrirait la voie à des applications novatrices et bénéfiques dans divers secteurs.
3. Développement de systèmes multi-agents plus complexes : Les recherches futures pourraient se concentrer sur le développement de systèmes multi-agents plus complexes et interconnectés, capables de collaborer de manière plus étroite et efficace dans des environnements dynamiques et imprévisibles.

Pour répondre encore plus précisément aux défis de la gestion du trafic, une perspective intéressante serait d'explorer comment notre système pourrait s'adapter de manière intelligente aux événements spéciaux. En tenant compte de ces occasions particulières, notre objectif est d'optimiser la fluidité de la circulation tout en minimisant les perturbations. Cette approche se concentre sur l'anticipation des changements de circulation, la gestion en temps réel pendant ces événements, et l'optimisation pour des situations spécifiques telles que des fermetures temporaires de routes ou la présence de véhicules prioritaires.

Bibliographie

[1] Abdalla, G. M., Abu-Rgheff, M. A., & Senouci, S. M. (2007). Current trends in vehicular ad hoc networks. *Ubiquitous Computing and Communication Journal*, 1-9.

[2] Abbas, F., Fan, P., & Khan, Z. (2018). A novel low-latency V2V resource allocation scheme based on cellular V2X communications. *IEEE Transactions on Intelligent Transportation Systems*, 20(6), 2185-2197.

[3] Abdoos, M., Mozayani, N., & Bazzan, A. L. (2011, October). Traffic light control in non-stationary environments based on multi agent Q-learning. *In Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on (pp. 1580-1585). IEEE.*

[4] Abowd GD, Dey AK, Brown PJ, Davies N, Smith M, Steggles P. Towards a better understanding of context and context-awareness. InHandheld and ubiquitous computing 1999 Sep 27 (pp. 304-307). *Springer Berlin Heidelberg.*

[5] Ali, K. A., Baala, O., & Caminada, A. (2011, May). Routing mechanisms analysis in vehicular city environment. *In Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd (pp. 1-5).IEEE.*

[6] Al-Sultan, S., Al-Doori, M. M., Al-Bayatti, A. H., & Zedan, H. (2014). A comprehensive survey on vehicular ad hoc network. *Journal of network and computer applications,* 37, 380-392.

[7] Ameur, A. I., Lakas, A., Bachir, Y. M., & Oubbati, O. S. (2022). Peer-to-peer overlay techniques for vehicular ad hoc networks : Survey and challenges. *Vehicular Communications*, 100455.

[8] Arel, I., Liu, C., Urbanik, T., & Kohls, A. (2010). Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems*, 4(2), 128-135.

[9] Baldauf M, Dustdar S, Rosenberg F. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*. 2007 Jan 1 ; 2(4) :263-77.

[10] Bauer, J., Kutsche, R. D., & Ehrmanntraut, R. (2003). Identification and modeling of contexts for different information scenarios in air traffic. *Technische Universität Berlin, Diplomarbeit.*

[11] Behlouli, N. B. (2006). *Ajout de mécanismes de réactivité au contexte dans les intergiciels pour composants dans le cadre d'utilisateurs nomades* (Doctoral dissertation, Université de Nice-Sophia Antipolis).

[12] Belbachir, A., El Fallah-Seghrouchni, A., Casals, A., & Pasin, M. (2019). Smart mobility using multi-agent system. *Procedia Computer Science,* 151, 447-454.

[13] Bertolini, D., Novikau, A., Susi, A., & Perini, A. (2006). TAOM4E : an Eclipse ready tool for Agent-Oriented Modeling. *Issue on the development process. Fondazione Bruno Kessler-irst. Reporte Técnico.*

[14] Boeglen, H., Hilt, B., Lorenz, P., Ledy, J., Poussard, A. M., & Vauzelle, R. (2011, January). A survey of V2V channel modeling for VANET simulations. *In 2011 Eighth International Conference on Wireless On-Demand Network Systems and Services (pp. 117-123). IEEE.*

[15] Bonner, A. J., & Kifer, M. (1995). Transaction logic programming (or, a logic of procedural and declarative knowledge).

[16] Bravo, M., Reyes-Ortiz, J. A., Sánchez-Martínez, L., & Alcántara-Ramírez, R. A. (2017). Hybrid Architecture to Support Context-Aware Systems. *In Multi-agent Systems. IntechOpen*

[17] Cabac, L., Haustermann, M., & Mosteller, D. (2018). Software development with Petri nets and agents : Approach, frameworks and tool set. *Science of Computer Programming,* 157, 56-70.

[18] Civicioglu, P. (2012). Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. **Computers & Geoscience**s, 46, 229-247.

[19] Cetin, N., Burri, A., & Nagel, K. (2003). A large-scale agent-based traffic micro simulation based on queue model. *In In proceedings of swiss transport research conference (strc), monte verita, ch (pp. 3–4272).* Citeseer.

[20] Chen, B., Cheng, H. H., & Palen, J. (2004, January). Agent-based real-time computing and its applications in traffic detection and management systems. *In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Vol. 46970, pp. 543-552).

[21] Chen, B., Cheng, H. H., & Palen, J. (2009). Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems. *Transportation Research Part C : Emerging Technologies,* 17(1), 1-10.

[22] C. LPCHERT, H. HARTENSTEIN et M. MAUVE J. TIAN D. HERRAMANN. A Routing Strategy for Vehicular Ad Hoc Networks in City Environments. *In IEEE Intelligent Vehicles Symposium,* volume 4792, pages 156–161. IV2003 Columbus, OH, USA„ 2003.

[23] Cheikh, M. S., Liang, J., & Wang, W. (2019). A survey of security services, attacks, and applications for vehicular ad hoc networks (vanets). *Sensors*, 19(16), 3589.

[24] 'Eclipse.' Available at : https ://www.eclipse.org/ide/ Accessed on 2020-07-10.

[25] F.A.M Barika, *Vers un IDS Intelligent à base d'Agents Mobiles*. Une approche à base d'agents mobiles pour la sécurité des systèmes d'informations sur le web. Mémoire de DEA Université de Tunis 2003.

[26] Ferber, J. Les Systèmes Multi-Agents : V*ers une intelligence collective*. InterEdition, 1995.[Pages 31, 32, 38].

[27] Ferrando, S. P., & Onaindia, E. (2013). Context-aware multi-agent planning in intelligent environments. *Information Sciences*, 227, 22-42.

[28] Festag, A., Noecker, G., Strassberger, M., Lübke, A., Bochow, B., Torrent-Moreno, M. & Kunisch, J. (2008). 'NoW–network on wheels' : Project objectives, technology and Achievements.

[29] Guo, C., Li, D., Chen, X., & Zhang, G. (2022). An adaptive V2R communication strategy based on data delivery delay estimation in VANETs. *Vehicular Communications*, 34, 100444.

[30] [42] Gligoric, N., Dimcic, T., Drajic, D., Krco, S., & Chu, N. (2012, November). Application-layer security mechanism for M2M communication over SMS. *In 2012 20th Telecommunications Forum (TELFOR) (pp. 5-8). IEEE.*

[31] Han, S., & Youn, H. Y. (2012). Petri net-based context modeling for context-aware systems. *Artificial Intelligence Review*, 37(1), 43-67.

[32] Haiouni, H., & Maamri, R. (2014). Context-Aware Automatic Service Selection Mechanism for Ambient Intelligent Environments. *In Ambient Intelligence-Software and Applications (pp. 1-11). Springer, Cham. context-aware .*

[33] Hartenstein, H. & Laberteaux, K. (2009). *VANET vehicular applications and inter-networking technologies (Vol. 1)*. John Wiley & Sons.

[34] Henricksen K, Indulska J. Modelling and using imperfect context information. In Pervasive Computing and Communications Workshops, 2004. *Proceedings of the Second IEEE Annual Conference on 2004 Mar 14 (pp. 33-37). IEEE.*

[35] Henricksen K, Indulska J. Developing context-aware pervasive computing applications : Models and approach. *Pervasive and mobile computing.* 2006 Feb 28 ;2(1) :37-64.

[36] H. Prakken and G. Vreeswijk. Logics for Defeasible Argumentation. *Handbook of philosophical logic*, pp. 4 :218–319, 2002.

[37] Hunn, C. (2009). Employing a Java Expert System Shell for Intelligent Support in Exploratory Activities. *Biz Logic Solutions.*

[38] Ivanov, R., Weimer, J., & Lee, I. (2018, April). Context-aware detection in medical cyber-physical systems. *In 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS) (pp. 232-241). IEEE.*

[39] J. CHOI, S. JUNG, Y. KIM et M. YOO. A Fast and Efficient Handover Authentication Achieving Conditional Privacy in V2I Networks. *Proceedings of the 9th International Conference on Smart Spaces and Next Generation Wired/Wireless Networking and Second Conference.*

[40] Jensen, K. (2013). Colored Petri nets : basic concepts, analysis methods and practical use (Vol. 1). *Springer Science & Business Media.*

[41] Kerraoui, S., Kissoum, Y., Redjimi, M., & Saker, M. (2016). MATT : multi agents testing tool based nets within nets. *Journal of information and organizational sciences,* 40(2), 165-184.

[42] Kaur, M. (2012). Vehicular ad hoc networks. *Journal of Global Research in Computer Science*,3(3), 61-64.

[43] Kirsch-Pinheiro M, Villanova-Oliver M, Gensel J, Martin H. Context-aware filtering for collaborative web systems : adapting the awareness information to the user's context.

[44] Köhler, M., Moldt, D., & Rölke, H. (2001, June). Modelling the structure and behaviour of Petri net agents. In International Conference on Application and Theory of Petri Nets (pp. 224-241). *Springer*, Berlin, Heidelberg.

[45] Köhler, M., Moldt, D., & Rölke, H. (2003). Modelling mobility and mobile agents using nets within nets. In Applications and Theory of Petri Nets 2003 : 24th International Conference, ICATPN 2003 Eindhoven, The Netherlands, June 23–27, 2003 Proceedings 24 (pp. 121-139). *Springer Berlin Heidelberg.*

[46] Kokinov B. A dynamic approach to context modeling. InProceedings of the IJCAI-95 workshop on modeling context in knowledge representation and reasoning. *LAFORIA* 1995 (Vol. 95, No. 11, pp. 199-209).

[47] Krajzewicz, D., Erdmann, J., Behrisch, M., & Bieker, L. (2012). Recent development and applications of SUMO-Simulation of Urban Mobility. *International journal on advances in systems and measurements*, 5(3&4).

[48] Kristensen, L.M. ; Jørgensen, J.B. ; Jensen, K. Application of Coloured Petri Nets in System Development. In Lecture Notes in Computer Science ; *Springer : Berlin/Heidelberg,* Germany, 2003 ; pp. 626–685. [Google Scholar].

[49] Lamy, J. P. (2012). SysML, un langage modèle. *Journal Technologie,* 179, 32-48.

[50] Mameri, S., Kissoum, Y., & Mazouzi, S. (2022, September). Integrating Jess rules in Renew Tool for Smart Car Guidance. *In 2022 International Conference on Advanced Aspects of Software Engineering (ICAASE) (pp. 1-8). IEEE.*

[51] Mameri, S., Kissoum, Y., & Redjimi, M. (2023). Mobile agents-based modelling for the vehicular network congestion problem resolution. *International Journal of Simulation and Process Modelling,* 21(1), 1-13.

[52] Mandaric, K., Skocir, P., & Jezic, G. (2021). A Smart Lighting Context-Aware Model Based on Colored Petri Net. In Agents and Multi-Agent Systems : Technologies and Applications 2021 : Proceedings of 15th KES International Conference, KES-AMSTA 2021, June 2021 (pp. 155-163). *Springer Singapore.*

[53] Mari KORKEA-AHO : Context-aware applications survey, 2000. URL http://www.cse.tkk.fi/fi/opinnot/T-110.5190/2000/applications/context-aware. html.

[54] M. BURMESTER, E. MAGKOS et V. CHRISSI. Strengthening Privacy Protection in VANETs. IEEE International Conference on Wireless and Mobile Computing, *Networking and Communications (WIMOB)*, 2008. 19 52.

[55] Milojicic, D., Douglis, F., Wheeler, R. : Mobility : processes, computers, and agents.ACM (1999).

[56] Miraoui, Moeiz (2009). *Architecture logicielle pour l'informatique diffuse : modélisation du contexte et adaptation dynamique des services.* Thèse de doctorat électronique, Montréal, École de technologie supérieure.

[57] M. K¨ohler, D. Moldt, and H. R¨olke : Modeling the behaviour of Petri net agents. In J. M. Colom and M. Koutny, editors, Proceedings of the 22nd Conference on Application and Theory of Petri Nets, volume 2075 of LNCS, pp. 224–241, *Springer* 2001.

[58] Naiem, A., Reda, M., El-Beltagy, M., & El-Khodary, I. (2010, March). An agent based approach for modeling traffic flow. In Informatics and Systems (INFOS), 2010 The 7th International Conference on (pp. 1-6). *IEEE.*

[59] Olariu, S., Weigle, M.C. : Vehicular Networks : From Theory to Practice. Chapman & Hall/CRC, first edn. (2009,

[60] O. ZHAO et C. VADD. Les protocoles de routage dans les réseaux ad hoc. Rapport de stage, *Université de Reims - UFR Sciences*, 2006.

[61] Pascoe J. Adding generic contextual capabilities to wearable computers. InWearable Computers, 1998. *Digest of Papers. Second International Symposium on 1998 Oct 19 (pp. 92- 99). IEEE.*

[62] P. Biondi, Architecture expérimentale pour la détection d'intrusions dans un système informatique. *Thèse de doctorat*, 2001.

[63] Peterson, J.L. Petri nets. ACM Surv. 1997, 9, 223–252. *[Google Scholar] [CrossRef]*

[64] P. M. Merlin and D. J. Farber, "Recoverability of Communication Protocols - Implications of a Theoretical Study," *IEEE Transactions on Communications, vol. 24, no. 9, pp. 1036–1043, Sep. 1976.*

[65] Q. XU et D. JIANG. Design and analysis of highway safety communication protocol in 5.9 GHz dedicated short range communication spectrum. *In Vehicular Technology Conference*, volume 4, pages 2451–2455, 2003.

[66] Rajabi, B. A. M. (2017). Coevolution Framework to Support Object-oriented Model Changes Using Coloured Petri Net Patterns (Doctoral dissertation, *University of Malaya (Malaysia)).*

[67] Raibulet C. Facets of adaptivity. In Software Architecture 2008 Sep 29 (pp. 342-345).*Springer Berlin Heidelberg.*

[68] R. Ammour, S. Amari, L. Brenner, I. Demongodin, and D. Lefebvre, "Observer design for output synchronized Petri nets," in European Control Conference (ECC), *Netherlands*, June 29 - July 2, 2021.

[69] Renew the reference networkshop.WWW page at http ://renew.de/. Contains the documentation for Renew and an introduction to reference nets.

[70] Robertson, D.I., 1969. TRANSYT : a traffic network study tool. *Road Research Laboratory.*

[71] RudigerValk, Universit atHamburg,Vogt-Kolln-Str.30,D-22527Hamburg, ,Germany valk@informatik.uni-hamburg.de4-8.07/07/2023.

[72] Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., & Edwards, D. D. (2010). Artificial intelligence : a modern approach (Vol. 3). *Prentice hall Upper Saddle River.*

[73] Ryan N, Pascoe J, Morse D. Enhanced reality fieldwork : the context aware archaeological assistant. *Bar International Series. 1999 ;750 :269-74.*

[74] Sathiyaraj, R., & Bharathi, A. (2020). An efficient intelligent traffic light control and deviation system for traffic congestion avoidance using multi-agent system. *Transport*, 35(3), 327-335.

[75] Sadiqa , J., Shahbazi, Z., & Byun, Y. C. (2022). Improving the road and traffic control prediction based on fuzzy logic approach in multiple intersections. *Mathematics*, 10(16), 2832.

[76] Schilit BN, Theimer MM. Disseminating active map information to mobile hosts.Network, *IEEE.* 1994 Sep ;8(5) :22-32.

[77] Schilit B, Adams N, Want R. Context-aware computing applications. InMobile Computing Systems and Applications, 1994. WMCSA 1994. *First Workshop on 1994 Dec 8(pp. 85-90). IEEE.*

[78] Sharmaa, A., Ranab, S., & Kaliac, L. (2014). Performance Comparison of AODV and DSDV Routing Protocols in Mobile Ad Hoc Networks. *International Journal of Emerging Research in Management & Technology ISSN :* 2278, 9359.

[79] Soares, G., Macedo, J., Kokkinogenis, Z., & Rossetti, R. (2013, May). An integrated framework for multi-agent traffic simulation using sumo and jade. In SUMO2013, *The first SUMO user conference* (pp. 15-17).

[80] Souad, M., Faiza, B., & Nabil, H. (2020, November). Formal modeling IoT systems on the basis of biagents* and maude. In 2020 International Conference on Advanced Aspects of Software Engineering (ICAASE) (pp. 1-7). *IEEE.*

[81] S. PATHAC et U. SHRAWAKER. Secured Communication in Real Time VANET,". *In the Second International Conference on Emerging Trends in Engineering and Technology,* pages 1151–1155, 2009.

[82] Strassner J, van der Meer S, O'Sullivan D, Dobson S. The use of context-aware policies and ontologies to facilitate business-aware network management. *Journal of Network and Systems Management.* 2009 Sep 1 ;17(3) :255-84.

[83] Strang, T., & Linnhoff-Popien, C. (2004, September). A context modeling survey. *In Workshop Proceedings*.

[84] T. Chaari, thèse de3éme cycle, Institut National des Sciences Appliquées, *Lyon*, (France), 2007.

[85] Tolbi, B. (2020). Optimal Supervisory Control of Two Phases Intersection Using Hybrid-based Model Approach. *Journal of Engineering Science & Technology Review, 13(5).*

[86] Trillo, R., Ilarri, S., Mena, E. : Comparison and performance evaluation of mobile agent platforms. In : The Third International Conference on Autonomic and Autonomous Systems (ICAS). pp. 41–46. *IEEE Computer Society (2007) .*

[87] Uddin, I., Rakib, A., Ali, M., Mahfooz Ul Haque, H., & Uddin, A. (2023). A Semantic-Based Approach to Modelling Smart Indoor Kitchen Garden. *Mobile Networks and Applications,* 1-16.

[88] Urra, O., Ilarri, S., Delot, T., & Mena, E. (2010). Mobile agents in vehicular networks : Taking a first ride. In Advances in Practical Applications of Agents and Multiagent Systems (pp. 119-124). *Springer*, Berlin, Heidelberg.

[89] Urra, O., Ilarri, S., & Trillo-Lado, R. (2017). An approach driven by mobile agents for data management in vehicular networks. *Information Sciences*, 381, 55-77.

[90] Valk, R. (1998, June). Petri nets as token objects. In International Conference on Application and Theory of Petri Nets (pp. 1-24). Springer, Berlin, Heidelberg. doi.org/10.1007/3-540-69108-1-1.

[91] Verma, K., Hasbullah, H., & Kumar, A. (2013, February). An efficient defense method against UDP spoofed flooding traffic of denial of service (DoS) attacks in VANET. In Advance Computing Conference (IACC), 2013 *IEEE 3rd International (pp. 550-555). IEEE.*

[92] Wang, F. Y. (2006). Driving into the future with ITS. *IEEE Intelligent Systems,* 21(3), 94-95.

[93] El Ghayam, Y., Erradi, M. (2011, May). Distributed context management in collaborative environment. In *2011 11th Annual International Conference on New Technologies of Distributed Systems* (pp. 1-8). IEEE.

[94] Yuhara, N. & Tajima, J. (2006). Multi-driver agent-based traffic simulation systems for evaluating the effects of advanced driver assistance systems on road traffic accidents. *Cognition, Technology & Work*, 8(4), 283-300.

[95] Yves Demazeau, et al (25 auteurs) ; Principes et architecture des systèmes multi-agents ; *Hermes Science Publication*, 2001.